FACULTY OF APPLIED SCIENCES
UNIVERSITY OF WEST BOHEMIA
IN PILSEN

DEPARTMENT OF
COMPUTER SCIENCE
AND ENGINEERING

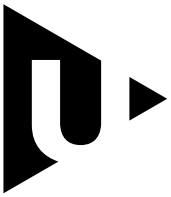
Master's Thesis

Use of predictive models in patients with salivary gland carcinomas in the Czech Salivary Gland Database application

Vojtěch Jelínek

PILSEN, CZECH REPUBLIC

2026



FACULTY OF APPLIED SCIENCES
UNIVERSITY OF WEST BOHEMIA
IN PILSEN

DEPARTMENT OF
COMPUTER SCIENCE
AND ENGINEERING

Master's Thesis

Use of predictive models in patients with salivary gland carcinomas in the Czech Salivary Gland Database application

Bc. Vojtěch Jelínek

Thesis advisor
Ing. Petr Brůha

PILSEN, CZECH REPUBLIC 2026

Citation in the bibliography/reference list:
JELÍNEK, Vojtěch. *Use of predictive models in patients with salivary gland carcinomas in the Czech Salivary Gland Database application*. Pilsen, Czech Republic, 2026. Master's Thesis. University of West Bohemia, Faculty of Applied Sciences, Department of Computer Science and Engineering. Thesis advisor Ing. Petr Brůha.

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta aplikovaných věd
Akademický rok: 2025/2026

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Bc. Vojtěch JELÍNEK**
Osobní číslo:	**A24N0105P**
Studijní program:	**N0613A140040 Softwarové a informační systémy**
Téma práce:	**Využití prediktivních modelů u pacientů s karcinomy slinných žláz v aplikaci Czech Salivary Gland Database**
Zadávající katedra:	**Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Prostudujte problematiku predikčních modelů v onkologii a porovnejte různé přístupy (včetně skórovacích systémů) s cílem vytvořit validní a dynamický model, který bude schopen se aktualizovat při přidávání nových dat do databáze.
2. V této souvislosti analyzujte strukturu aplikace Czech Salivary Gland Database vyvíjené na ZČU z hlediska potřeb predikčních modelů pro zpracování klinických, patologických a léčebných dat pacientů s karcinomy slinných žláz.
3. Ve spolupráci s vedoucím práce, lékaři z FN Motol a dle výsledků provedené rešerše navrhněte modely pro výpočet individuálního prognostického skóre pacientů.
4. Navržené modely integrujte do aplikace Czech Salivary Gland Database jako samostatný nástroj pro podporu klinického rozhodování a plánování léčby pacientů. Výsledné řešení nasaďte na pracovišti FN Motol a připravte dokumentaci pro jeho praktické využití.
5. Ověřte funkčnost a přesnost modelů na retrospektivních datech a testujte predikční přesnost zvolených modelů, výsledky kriticky zhodnoťte.

Rozsah diplomové práce:	**doporuč. 50 s. původního textu**
Rozsah grafických prací:	**dle potřeby**
Forma zpracování diplomové práce:	**tištěná/elektronická**
Jazyk zpracování:	**Angličtina**

Seznam doporučené literatury:

dodá vedoucí diplomové práce

Vedoucí diplomové práce:	**Ing. Petr Brůha** Katedra informatiky a výpočetní techniky
Datum zadání diplomové práce:	**10. září 2025**
Termín odevzdání diplomové práce:	**14. května 2026**

L.S.

doc. Ing. Miloš Železný, Ph.D.	**doc. Ing. Přemysl Brada, MSc., Ph.D.**
děkan	vedoucí katedry

V Plzni dne 30. září 2025

Declaration

I hereby declare that this Master's Thesis is completely my own work and that I used only the cited sources, literature, and other resources. This thesis has not been used to obtain another or the same academic degree.

I further declare that in preparing this work, I used Gemini 3.1 Pro and Grammarly for grammar and style checking. After using these tools, I carefully reviewed and edited the content as needed and assume full responsibility for the final content of the work.

I acknowledge that my thesis is subject to the rights and obligations arising from Act No. 121/2000 Coll., the Copyright Act as amended, in particular the fact that the University of West Bohemia has the right to conclude a licence agreement for the use of this thesis as a school work pursuant to Section 60(1) of the Copyright Act.

V Plzni, on 14 May 2026

. . . . . .

Vojtěch Jelínek

Abstract

Salivary gland carcinomas are rare, heterogeneous malignancies that complicate prognostic stratification. This thesis documents the transformation of the Czech Salivary Gland Database from a retrospective repository into a dynamic clinical decision support system. The system uses a Python-based sidecar architecture integrated with an offline Electron/React/SQLite stack to ensure GDPR compliance and data sovereignty at the workstation level.

We implemented two predictive paradigms: a semi-parametric Cox Proportional Hazards baseline and a non-parametric Random Survival Forest. Validation on $N = 122$ patients via the .632 bootstrap estimator yielded a concordance index of 0.814 for the Random Survival Forest, although overlapping confidence intervals reflect the constraints of the limited sample size. The application, featuring an explainability layer for clinical trust, is deployed at Motol University Hospital to assist with individualized treatment planning.

Abstrakt

Karcinomy slinných žláz jsou vzácná a heterogenní onemocnění komplikující prognostickou stratifikaci. Tato práce dokumentuje transformaci aplikace Czech Salivary Gland Database z retrospektivního repozitáře na dynamický systém pro podporu klinického rozhodování. Architektura využívá Python sidecar v rámci offline prostředí Electron/React/SQLite, čímž zajišťuje soulad s GDPR a datovou suverenitu na úrovni pracovní stanice.

Implementovány byly dva predikční modely: Coxův model proporcionálních rizik a Random Survival Forest. Validace na souboru $N = 122$ pacientů pomocí .632 bootstrap estimátoru vykázala u modelu Random Survival Forest index konkordance 0,814, přičemž překrývající se intervaly spolehlivosti reflektují statistické limity rozsahu dat. Aplikace, obsahující vrstvu pro vysvětlitelnost výsledků, je nasazena ve Fakultní nemocnici v Motole pro účely individualizovaného plánování léčby.

Keywords

Salivary gland carcinoma • Survival analysis • Random Survival Forest • Prognostic score • Czech Salivary Gland Database

Acknowledgement

I would like to express my sincere gratitude to my supervisor, Ing. Petr Brůha, and to MUDr. David Kalfeřt, Ph.D., for their invaluable guidance and support throughout the development of this Master's thesis. I am also deeply grateful to my family and my girlfriend for their unwavering encouragement, patience, and understanding.

Contents

Introduction 1

Salivary gland carcinomas are very rare neoplasms, accounting for approximately 5% of all head and neck malignancies in Europe [1]. Worldwide annual incidence rates vary between less than 2 and greater than 0.05 per 100,000 individuals [2]. Beyond its rarity, salivary gland carcinomas are extremely heterogeneous, with current classifications recognizing 21 distinct histological types [3, 4, 1, 5].

The Czech Salivary Gland Database was initially established as a semester project and subsequently expanded into a bachelor's thesis focused on collecting patient clinical data and providing survival analysis using Kaplan-Meier curves [6]. As clinicians at Motol University Hospital accumulated a significant number of entries, the potential to use this comprehensive information for predictive modeling became evident. Our preliminary research suggested that further extensions to the platform should incorporate machine learning for individualized predictions [7].

In this thesis, we address the lack of available software that allows clinicians to locally accumulate data and train custom models based on their own data. The application provides a prognostic score for clinical decision-making and is deployable without internet connectivity, ensuring data security while enabling anonymized exports for workspace collaboration. Our primary goal was to develop a dynamic framework featuring a Cox Proportional Hazards baseline and a machine learning model that updates with new data entries.

Based on a comparison of approaches for our sparse dataset and consultations with FN Motol experts, we selected Random Survival Forests as the machine learning algorithm. The engine uses a sidecar architecture pattern, with communication via JSON over standard input. To mitigate the low event-per-variable ratio of the small dataset, we utilize the .632 bootstrapping method for validation to obtain more reliable estimates.

The work progresses through a theoretical review in Chapter 2 and an analysis of the legacy implementation in Chapter 3. Chapter 4 details the system redesign and the proposed methodology for the machine learning models, and Chapter 5 describes the new implementation. Finally, Chapter 6 provides the statistical validation of the resulting models.

1.1 State of the Art

The current landscape of head and neck oncology data management is defined by a tiered hierarchy of registries, which were audited and summarized in our previous work [7]. Salivary gland tumors are relatively rare malignancies that require specialized care and robust, high-granularity data [7], yet existing medical databases and general-purpose Electronic Health Record systems are often ill-suited to managing the detailed, specific data required for this patient population [7]. Conventional data systems, such as general cancer registries, frequently lack the specific clinical fields and integrated tools for survival analysis or detailed cohort tracking necessary to identify prognostic indicators [7]. Our paper identified three primary international frameworks:

- **Epidemiological Registries:** Programs such as the SEER provide large population level datasets [8, 9]. However, these registries often lack the integrated clinicopathological depth required for everyday clinical practice [7].
- **Pathology-Centric Standards:** The ICCR provides evidence-based pathology datasets designed for global standardization [10, 11]. While robust, these frameworks are primarily intended for pathologists and frequently miss complete clinical trajectories [7].
- **Specialized National Databases:** The DAHANCA represents a global gold standard for national cancer registries [12]. It demonstrates how transitioning from localized institutional tracking to a national database can improve clinical outcomes [7, 12].

Predictive Modelling in Clinical Oncology — 2

This chapter establishes the theoretical and mathematical foundation for survival analysis as applied to clinical oncology. It begins by defining the core principles of time-to-event data, focusing on the statistical treatment of right-censored observations and the derivation of the survival and hazard functions.

The text subsequently analyzes various modeling paradigms, beginning with parametric frameworks. Within the AFT model, the roles of specific distributions are evaluated based on their ability to represent different hazard rate profiles. After this, we examine the semi-parametric Cox Proportional Hazards model.

To relate statistical theory to established medical practice, we review standardized clinical scoring systems.

In the final sections, we explore machine learning methodologies, emphasizing ensemble learning through Random Survival Forests and boosting algorithms (AdaBoost and Gradient Boosting). Additionally, we address the application of deep feed-forward neural networks in survival analysis via the DeepSurv architecture. Finally, we conclude with a formal definition of evaluation metrics used to assess model performance.

2.1 Survival Principles

Survival analysis is the analysis of time-to-event data [13]. In oncology, these events are typically death (Overall Survival) or tumor recurrence (Recurrence-Free Survival). Unlike standard regression or classification tasks, the design of survival analysis handles the temporal nature of the data and the statistical challenges posed by incomplete observations. The distribution of the survival time, denoted by the non-negative random variable T, is fundamentally described by two mathematically linked functions: the survival function and the cumulative hazard function.

The survival function is defined as

$$S(t) \equiv 1 - F(t) = P(T > t) = \int_t^\infty f(x)dx \tag{2.1}$$

where $F(t)$ is cumulative distribution function, $P(T > t)$ probability function and $f(x)$ probability density function [14].

While the survival function provides a cumulative perspective on the probability of remaining event-free, the hazard rate function, $h(t)$, focuses on the instantaneous risk. The hazard represents the probability that an event occurs within an infinitely small time interval Δt, given that the individual has survived up to time t. This relationship is expressed as

$$h(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T < t + \Delta t \mid T \geq t)}{\Delta t} = \frac{f(t)}{1 - F(t)} = \frac{f(t)}{S(t)} \tag{2.2}$$

When integrating over the hazard rate function we get the cumulative hazard function expressed as

$$H(t) = \int_0^t h(u)du = \int_0^t \frac{f(u)}{1 - F(u)} = -\log[1 - F(u)]\Big|_0^t = -\log S(t) \tag{2.3}$$

Finally using the cumulative hazard function we get the expression that connects the survival with the hazard [14, 15].

$$H(t) = -\ln(S(t)) \quad \Big/ \cdot(-1) \tag{2.4}$$

$$-H(t) = \ln(S(t)) \quad \Big/ \exp(\dots) \tag{2.5}$$

$$e^{-H(t)} = e^{\ln(S(t))} \tag{2.6}$$

$$e^{-H(t)} = S(t) \tag{2.7}$$

$$S(t) = e^{-H(t)} \tag{2.8}$$

Where this relation essentialy says the higher the hazard, the lower the survival [13].

The primary distinction between classical statistics and survival analysis is the presence of censoring, which occurs when the exact time of the event is unknown. Although survival frameworks can accommodate left-censoring ($T \leq t$) and interval-censoring ($T \in [L, R]$), clinical databases like the CSGDB primarily encounter right-censoring. This occurs when the event of interest remains unobserved at the study's end, or a patient drops out of follow-up ($T > t$).

To estimate the model coefficients in the presence of such data, we construct a likelihood function L. For a sample of n independent individuals, the likelihood contribution is determined by an indicator variable δ_i:

$$L = \prod_{i=1}^{n} [f_i(t_i)]^{\delta_i} [S_i(t_i)]^{1-\delta_i} \tag{2.9}$$

$$\text{where } \delta_i = \begin{cases} 1 & \text{if event observed} \\ 0 & \text{if right-censored} \end{cases}$$

By substituting the relationship $f_i(t_i) = h_i(t_i)S_i(t_i)$, we can simplify the likelihood into a form that separates the instantaneous risk from the cumulative survival:

$$\text{Substituting } f_i(t_i) = h_i(t_i)S_i(t_i) :$$

$$L = \prod_{i=1}^{n} [h_i(t_i)S_i(t_i)]^{\delta_i} [S_i(t_i)]^{1-\delta_i} \tag{2.10}$$

$$L = \prod_{i=1}^{n} h_i(t_i)^{\delta_i} S_i(t_i)^{\delta_i} S_i(t_i)^{1-\delta_i} \tag{2.11}$$

$$L = \prod_{i=1}^{n} h_i(t_i)^{\delta_i} S_i(t_i)^{\delta_i+1-\delta_i} \tag{2.12}$$

$$L = \prod_{i=1}^{n} h_i(t_i)^{\delta_i} S_i(t_i) \tag{2.13}$$

In this formulation, f_i, S_i, and h_i denote the density, survival, and hazard functions evaluated for the i-th individual, implicitly incorporating their observed clinical characteristics and the model parameters [14]. This final expression demonstrates that every individual contributes their survival information $S_i(t_i)$ to the model, while only those experiencing an event ($\delta_i = 1$) contribute to the hazard component $h_i(t_i)$ [16]. This allows all patients to be included in the analysis, regardless of whether their specific outcome was observed.

2.2 Parametric Models

Parametric survival models assume that the survival time T follows a specific, known probability distribution [17, 13]. By assuming a fixed functional form for the survival or hazard functions, the model parameters can be estimated directly from the data. These models are highly efficient when the distributional assumptions hold, providing a complete description of the survival process across the entire time horizon [17].

2.2.1 Accelerated Failure Time Model

The AFT model is a parametric framework that assumes a linear relationship between the logarithm of the survival time T and a vector of clinical characteristics X [14]. The AFT is expressed

$$\ln(T) = \beta X + z, \tag{2.14}$$

where β is a vector of parameters, z is an error term and X is a vector of individual characteristics. This can be rewritten as a generalized linear model in the form

$$Y = \mu + \sigma u, \tag{2.15}$$

where $Y \equiv \ln(T)$, $\mu \equiv \beta X$, and $u = z/\sigma$ is an error term with a density function $f(u)$ and a scale factor σ. In this context, the distributional assumptions regarding the error term u determine the specific functional form of the survival model (e.g., Weibull or Lognormal) [14].

The "Accelerated Failure Time" label originates from how the covariates influence the survival process. By defining a time-scaling factor:

$$\psi = \exp(-\beta X) = \exp(-\mu), \tag{2.16}$$

the survivor function $S(t, X)$ for a specific individual can be expressed in terms of a baseline survivor function $S_0(t)$. Starting from the fundamental definition:

$$S(t, X) = P(T > t \mid X), \tag{2.17}$$

and applying the logarithmic transformation $Y = \ln T = \mu + \sigma u$, we obtain:

$$\begin{aligned} S(t, X) &= P(Y > \ln t \mid X) \\ &= P(\mu + \sigma u > \ln t) \\ &= P(\sigma u > \ln t - \mu) \\ &= P(\exp(\sigma u) > t \exp(-\mu)). \end{aligned} \tag{2.18}$$

The baseline survivor function $S_0(t)$ is defined at the point where all covariates are zero ($X = 0$, implying $\mu = 0$):

$$\begin{aligned} S_0(t) &= P(T > t \mid X = 0) \\ &= P(\exp(\sigma u) > t \exp(0)) \\ &= P(\exp(\sigma u) > t). \end{aligned} \tag{2.19}$$

By comparing Equation 2.18 with the baseline form in Equation 2.19, it is evident that:

$$S(t, X) = S_0(t \exp(-\mu)) = S_0(t\psi). \tag{2.20}$$

The term ψ acts as a constant scaling factor on the time axis, effectively altering the "speed" of the internal clinical clock. Jenkins [14] identifies two key scenarios for this property:

- $\psi > 1$ **(Acceleration):** If $\exp(-\mu) > 1$ (where $\beta X < 0$), the internal clock ticks faster. The failure is accelerated, and the survival time is shortened.
- $\psi < 1$ **(Deceleration):** If $\exp(-\mu) < 1$ (where $\beta X > 0$), the internal clock ticks slower. The failure is decelerated, and the survival time is lengthened.

An intuitive illustration of this scaling is the comparison of longevity between species; if one calendar year for a dog is equivalent to seven years for a human, the dog ages faster relative to the human baseline, representing an acceleration factor of $\psi = 7$ [14].

2.2.2 Distributions

The choice of the probability distribution for the error term u in Eq. 2.15 directly dictates the functional form of the survival model for T. In clinical practice, the selection of a specific distribution is typically guided by the hypothesized shape of the hazard rate function $h(t)$.

Exponential Distribution

The Exponential distribution is the simplest parametric case, where the scale factor σ is fixed at 1 and the error term u follows a one-parameter extreme value distribution. This model is characterized by a constant hazard rate $h(t) = \lambda$, implying that the probability of the event occurring in any time interval is independent of how much time has already passed [15]. The survival function is defined as

$$S(t) = e^{-\lambda t} \tag{2.21}$$

While mathematically elegant due to its "memoryless" property [18, 19], it is rarely suitable for oncology datasets, as the biological risk of recurrence or death usually varies significantly over the patient's clinical history [20].

Weibull Distribution

The Weibull distribution is the most widely used parametric model in medical research, primarily due to its flexibility in allowing the shape parameter α (where the scale factor of the error term $\sigma = 1/\alpha$) to be a free parameter [14]. This allows the hazard rate $h(t)$ to be monotonic—either strictly increasing or strictly decreasing over time [17]. The hazard rate is defined as:

$$h(t) = \alpha\lambda t^{\alpha-1}, \tag{2.22}$$

Moreover, the corresponding survival function is given by:

$$S(t) = e^{-\lambda t^{\alpha}}. \tag{2.23}$$

When $\alpha = 1$, the Weibull model reduces to the Exponential model with a constant hazard rate. A unique theoretical advantage of the Weibull distribution is that it is the only distribution that satisfies the assumptions of both the Accelerated Failure Time and Proportional Hazards frameworks [14]. Despite its versatility and widespread use in the statistical literature, the Weibull model is often not the primary choice for analyzing complex clinical data, as its requirement for a monotonic hazard may not capture the fluctuating risk profiles often observed in oncological cohorts [21].

Log-normal Distribution

In the Log-normal model, the error term u follows a standard normal distribution, $u \sim N(0, 1)$. This results in a specification where the log-survival time $Y = \ln T$ is linearly related to the covariates X, mirroring the structural form of OLS regression. However, unlike OLS, the Log-normal survival model uses maximum likelihood estimation to incorporate information from right-censored observations, as discussed in Section 2.1 [14]. The survivor function is defined as:

$$S(t) = 1 - \Phi\left(\frac{\ln t - \mu}{\sigma}\right), \tag{2.24}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution, $\mu = X\beta$ is the location parameter, and σ is the scale parameter.

The Log-normal hazard function is notably non-monotonic: it initially increases from zero to a peak, then declines, eventually approaching zero as $t \to \infty$ [22]. This risk profile is particularly valuable for modeling clinical events with a distinct peak in incidence. In oncological contexts, this may represent specific recurrence windows during the first few years post-surgery, when the risk is highest, before subsiding in long-term survivors, or the transient risk of mortality following high-risk surgical procedures [19].

Log-logistic Distribution

The Log-logistic distribution is defined by a logistic distribution of the error term u, where the density is given by $f(u) = \frac{\exp(u)}{(1+\exp(u))^2}$ [14]. A significant mathematical advantage of the Log-logistic model over the Log-normal is that its survivor and hazard functions exist in closed form, requiring no numerical integration. The survivor function is defined as:

$$S(t) = \frac{1}{1 + (\lambda t)^p}, \tag{2.25}$$

where $\lambda = \exp(-\mu)$ is the scale parameter and $p = 1/\sigma$ is the shape parameter. This parameterization links directly to the AFT framework, where $\mu = X\beta$. Similar to the Log-normal model, the Log-logistic hazard function can capture non-monotonic risk profiles when $p > 1$:

$$h(t) = \frac{\lambda p(\lambda t)^{p-1}}{1 + (\lambda t)^p}. \tag{2.26}$$

The Log-logistic hazard has "heavier tails" than the Log-normal distribution. In practical terms, this means the hazard rate declines toward zero more slowly as $t \to \infty$. This property is particularly well-suited for clinical scenarios where the risk is expected to rise initially and then taper off slowly, maintaining a persistent, albeit declining, risk for long-term survivors [17].

2.2.3 MLE Estimation

MLE is a standard approach in statistical modeling used to identify the probability distribution most likely to have generated a specific data sample [23]. The process begins by defining a model as a family of probability distributions indexed by a parameter vector θ [23]. While a probability density function $f(t, X \mid \theta)$ specifies the probability of observing data given a set of parameters, the likelihood function $L(\theta \mid t, X)$ reverses this relationship to focus on the parameters given the observed data [23]. For n independent observations, the joint likelihood is the product of individual density functions [23]:

$$L(\theta \mid t_i, X_i, \delta_i) = \prod_{i=1}^{n} h(t_i, X_i)^{\delta_i} S(t_i, X_i). \tag{2.27}$$

For computational convenience, the estimate θ_{MLE} is obtained by maximizing the log-likelihood function, $l(\theta) = \ln L(\theta)$ [23]. Because these two functions are monotonically related, they share the same maximum point [23]. This transformation simplifies the calculation by converting the product into a summation [23]:

$$l(\theta) = \sum_{i=1}^{n} \left[\delta_i \ln h(t_i, X_i) + \ln S(t_i, X_i)\right]. \tag{2.28}$$

If the log-likelihood is differentiable, the resulting estimator must satisfy the likelihood equations where the first partial derivatives vanish [23]. In practice, because many survival models involve highly non-linear densities, these estimates are typically sought numerically using iterative optimization algorithms [23]. While this heuristic process can be subject to the local maxima problem—where an algorithm converges on a sub-optimal solution—it is the preferred method due to its optimal properties [23]. These include sufficiency, in which the estimator captures all information about the parameter contained in the data; consistency, which ensures the true parameter value is recovered as the sample size increases; and efficiency, which minimizes the variance [23]. By maximizing this log-likelihood, the parametric distributions are rigorously fitted to the patient cohort's survival times and censoring indicators.

2.3 Semi-Parametric Models

A semi-parametric model is defined by the partitioning of the parameter θ into a finite-dimensional parametric component β and an infinite-dimensional nonparametric component η [24, 22]. The parametric component β is a vector of q dimensions of coefficients that describes the specific effects of the clinical covariates on the hazard rate. In contrast to strictly parametric models where the entire

parameter space Ω is finite-dimensional, the semi-parametric framework treats the baseline hazard as an infinite-dimensional nuisance parameter η [24]. This allows for the estimation of relative risks, or hazard ratios, without assuming a specific underlying distribution for survival times [22].

2.3.1 The Cox PH Model

The Cox proportional hazards model, also known as the Cox regression model, is defined by the following form:

$$h(t, X) = h_0(t) \exp(X\beta), \tag{2.29}$$

where $X\beta$ represents the dot product between the vector of explanatory variables X and the finite-dimensional parameter vector β. The function $h_0(t)$ denotes the baseline hazard function, which represents the hazard for an individual when the covariate vector $X = 0$ [22, 25]. The term $\exp(X\beta)$ represents the hazard ratio (or relative hazard) for an individual with a specific set of explanatory variables X, relative to a baseline individual for whom $X = 0$. Since this ratio is independent of time, it implies that the hazard functions for different individuals are proportional to one another over the entire duration of follow-up [22].

2.3.2 Baseline Hazard

The baseline hazard represents the risk of the event occurring at time t for an individual with all covariates set to zero ($X = 0$) [25]. In the semi-parametric framework, we leave this component unspecified, meaning the model makes no assumptions about the shape of the underlying distribution of survival times [22]. While it is treated as a nuisance parameter during the estimation of the regression coefficients β, the baseline hazard serves as the essential component for prediction. It provides the starting point for calculating individualized survival curves.

2.3.3 PH Assumption

The primary requirement of the Cox model is the PH assumption. This assumption dictates that the hazard ratio comparing any two individuals is constant over time. Mathematically, this implies that the effect of the explanatory variables X is multiplicative and does not interact with time t. As demonstrated in Equation 2.29, because the term $\exp(X\beta)$ contains no temporal component, the ratio of the hazards for two different patient profiles remains fixed:

$$\frac{h(t, X_i)}{h(t, X_j)} = \frac{h_0(t) \exp(X_i\beta)}{h_0(t) \exp(X_j\beta)} = \exp[\beta(X_i - X_j)]. \tag{2.30}$$

In this ratio, the baseline hazard $h_0(t)$ cancels out, leaving a result that is independent of time. For the model to serve as a valid predictive tool, this proportionality must hold. If the effect of a covariate changes as time progresses, the estimated coefficients β would only represent an average effect, potentially leading to biased predictions. Consequently, verifying the PH assumption—typically through the analysis of Schoenfeld residuals—is a standard diagnostic step to ensure the model's generalizability and accuracy [22].

2.4 Scoring Systems

While the statistical models we discussed in the previous sections provide the mathematical foundation for survival analysis, they are often limited in clinical practice due to their complexity. Scoring systems serve as the bridge between theoretical models and clinical practice, simplifying multivariable equations into intuitive tools. These systems enable clinicians to estimate a patient's prognosis, guide treatment decisions, and standardize risk communication across the medical community.

2.4.1 TNM Classification

The TNM system serves as an international standard for cancer classification [26]. TNM focuses on classification by anatomical extent of disease as determined clinically and histopathologically [26]. The classification is composed of three components:

- **T** – extent of the primary tumour,
- **N** – absence or presence and extent of regional lymph node metastasis,
- **M** – absence or presence of distant metastasis [26].

The application of the system is governed by six general rules to ensure consistency in medical documentation and research [26]:

1. Microscopical confirmation and separated reports.
2. Both clinical and pathological classifications are described.
3. T, N, M groups can be grouped into stages, where once the stages are established, they must remain unchanged in the medical records.
4. When there is uncertainty on the correct T, N, M category, the lower category should be chosen.

5. In case of multiple primary tumors, the one with the highest T should be classified, and the number of tumors or multiplicity should be indicated in parentheses (eg, T2(4)).

6. The TNM categories and stages can be expanded for clinical or research purposes, while the recommended definitions are not changed.

The currently used version for salivary glands is AJCC Version 9, for which the staging calculations are shown in Table 2.1 [27].

Table 2.1: AJCC Version 9 Staging for Salivary Gland Tumors.

Stage	T Category	N Category	M Category
Stage 0	Tis	N0	M0
Stage I	T1	N0	M0
Stage II	T2	N0	M0
Stage IIIA	T3, T4	N0	M0
	T1, T2	N1	M0
Stage IIIB	T1, T2	N2	M0
	T3, T4	N1, N2	M0
Stage IV	Any T	Any N	M1

2.4.2 The Milan System

The MSRSGC is a diagnostic framework designed to standardize communication between clinicians and pathologists by providing a tiered approach to reporting FNA results of salivary gland lesions [28]. The system utilizes six diagnostic categories, each associated with a specific ROM [28]. This risk-stratification model enhances clinical decision-making by directly correlating cytologic findings with management strategies [28].

The six diagnostic categories of the MSRSGC, along with their respective descriptions and associated risks of malignancy (ROM), are summarized in Table 2.2.

2.5 Machine Learning

The application of machine learning in survival analysis enables the discovery of non-linear prognostic relationships that traditional statistical models may fail to capture. This section details ensemble learning, which integrates multiple individual learners to achieve generalization performance through parallel or sequential architectures. Parallel methodologies, exemplified by the Random Survival Forest, reduce variance by growing trees independently on bootstrap samples and using

Table 2.2: The Milan System (MSRSGC).

Category	Description	ROM
I. Non-Diagnostic	Insufficient material for cytologic diagnosis.	15%
II. Non-Neoplastic	Benign entities (e.g., inflammation, infection).	11%
III. AUS	Limited atypia; indeterminate for neoplasm.	30%
IVA. Neoplasm: Benign	Established benign neoplasms.	<3%
IVB. Neoplasm: SUMP	Neoplasm present; malignancy cannot be excluded.	35%
V. Suspicious	Highly suggestive but not unequivocal for malignancy.	83%
VI. Malignant	Diagnostic of malignancy.	98%

survival-specific splitting criteria. Contrarily, sequential methodologies, such as AdaBoost and Gradient Boosting, focus on reducing bias by iteratively constructing strong learners that correct the errors of their predecessors. Finally, we explore artificial neural networks, specifically the state-of-the-art DeepSurv framework, which leverages deep feed-forward architectures to model high-level interactions between patient covariates and provides personalized treatment recommendations.

2.5.1 Ensemble Learning

Ensemble learning is a committee-based approach that trains and integrates multiple learners to address a single predictive task. The typical workflow of an ensemble system involves training a set of individual learners, followed by their combination using specific aggregation strategies. These individual models are typically generated using learning algorithms, such as decision trees or neural networks. An ensemble is homogeneous when all the learners are of the same type, i.e., base learners generated by a base learning algorithm. In contrast, a heterogeneous ensemble integrates multiple component learners [29].

The primary objective of this methodology is to achieve a generalization performance that is more robust than that of any individual learner. While theoretical foundations often emphasize the weak learner hypothesis, practical applications frequently employ strong learners to enhance computational efficiency and leverage domain-specific knowledge. Mathematically, the efficacy of an ensemble is predicated on the dual requirements of accuracy and diversity. For the ensemble to succeed, the learners must maintain a high degree of predictive accuracy while remaining sufficiently different from one another so that their predictive errors do not coincide [29].

To address the challenge of generating accurate and diverse learners, current methodologies are broadly divided into two categories. Sequential methods, such as Boosting, generate learners in a dependent manner to reduce bias by iteratively correcting preceding errors. Conversely, parallel methods, such as Bagging and

Random Forests, generate learners independently to reduce variance by averaging decorrelated models [29].

2.5.2 Random Survival Forest

The RSF is a specialized extension of the RF ensemble paradigm, specifically engineered for analyzing right-censored survival data. While traditional RF implementations are optimized for classification and regression, RSF adapts the randomization logic to accommodate the complexities of time-to-event outcomes, namely censoring and the non-normal distribution of survival times [30]. The algorithm adheres to the parallel ensemble framework by introducing randomness through two primary mechanisms: using independent bootstrap samples to grow each tree and selecting a random subset of candidate variables at each node for splitting.

The fundamental logic of RSF departs from standard trees in its splitting criterion. Rather than minimizing node impurity through Gini or entropy measures, RSF utilizes a survival-specific criterion (typically the log-rank test statistic) to maximize the survival difference between daughter nodes. This process effectively pushes dissimilar cases apart, ensuring that individuals with homogeneous survival profiles are assigned to terminal nodes. By recursively partitioning the data in this manner, the model automatically uncovers non-linear effects and high-order interactions between clinical and pathological variables without the need for manual feature transformation or the proportional hazards assumption required by the semi-parametric Cox model [30].

The predictive power of the forest is derived from the terminal nodes of its constituent trees. A tree continues to grow until a saturation point is reached, defined by a minimum threshold of d_0 unique deaths in every terminal node [30]. For each terminal node h, the CHF is estimated using the Nelson-Aalen estimator:

$$\hat{H}_h(t) = \sum_{t_{l,h} \leq t} \frac{d_{l,h}}{Y_{l,h}} \tag{2.31}$$

where $d_{l,h}$ represents the number of deaths and $Y_{l,h}$ denotes the individuals at risk at time $t_{l,h}$. To determine the prognosis for a specific patient with covariates x, the case is dropped down each tree until it falls into a unique terminal node [30]. The ensemble CHF, denoted as $H_e^*(t|x)$, is then calculated by averaging the Nelson-Aalen estimators from all B trees in the forest:

$$H_e^*(t|x) = \frac{1}{B} \sum_{b=1}^{B} H_b^*(t|x) \tag{2.32}$$

This aggregation reduces the variance inherent in individual survival trees, leading to a more stable and accurate prognostic signature. To evaluate the predictive performance of this ensemble, RSF typically employs Harrell's C-index.

2.5.3 Boosting Models

Boosting is a family of ensemble algorithms that transform a collection of weak learners into a single, robust strong learner through a sequential, iterative process. This process begins with training an initial base learner, after which the distribution of training samples is adjusted based on that learner's predictive performance. Specifically, the algorithm reweights the training set so that instances misidentified or incorrectly classified by the preceding model receive increased focus in subsequent iterations. This sequential training is repeated until the ensemble reaches a predefined number of base learners, T. At this point, the trained models are combined using a weighted average to produce the final predictive output [29].

AdaBoost

AdaBoost [31] is a foundational algorithm within the boosting family that constructs a strong learner by iteratively aggregating a series of weak learners. The model is formulated as an additive combination of base learners $h_t(x)$, where each learner is assigned a weight α_t determined by its specific error rate ϵ_t on the training distribution [29]. The final ensemble $H(x)$ is defined by the following additive model:

$$H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{2.33}$$

The optimization objective of AdaBoost is the minimization of the exponential loss function, $\mathbb{E}_{x\sim D}[e^{-f(x)H(x)}]$. As a continuously differentiable surrogate for the 0/1 loss, the exponential loss ensures that the ensemble minimizes the classification error rate and eventually approaches the Bayes-optimal error.

The algorithm functions by iteratively re-weighting the training sample distribution D_t. After each training round, the importance weight of the current base learner is calculated as:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right) \tag{2.34}$$

Samples misclassified by h_t are assigned increased weights in the subsequent distribution D_{t+1}, forcing the next learner to focus on the most challenging cases. From the perspective of bias-variance decomposition, this sequential mechanism primarily reduces model bias.

Gradient Boosting

GBM, as originally proposed by Friedman [32], frame the problem of supervised learning as a function estimation task. The primary objective is to reconstruct an unknown functional dependence $f : x \rightarrow y$ from a provided dataset $\{(x_i, y_i)\}_{i=1}^{N}$ [33]. This estimation is achieved by identifying a function $\hat{f}(x)$ that minimizes a specified loss function $\Psi(y, f)$, representing a numerical optimization in function space:

$$\hat{f}(x) = \arg\min_{f(x)} \Psi(y, f) \tag{2.35}$$

The algorithm operates through a sequential, iterative process in which a model is constructed stagewise. Starting with an initial base learner, the ensemble is updated in each iteration t by adding a new functional increment, $h(x, \theta_t)$ [33], such that the model is updated as follows:

$$\hat{f}_t(x) \leftarrow \hat{f}_{t-1}(x) + \rho_t h(x, \theta_t) \tag{2.36}$$

Because direct parameter estimation for the increment can be computationally prohibitive for arbitrary loss functions or complex base-learners, the algorithm identifies a new function most parallel to the negative gradient $\{g_t(x_i)\}_{i=1}^{N}$ along the observed data [33]. These gradients serve as pseudo-residuals and are calculated by taking the derivative of the loss function with respect to the current model's predictions:

$$g_t(x_i) = \left[\frac{\partial \Psi(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=\hat{f}_{t-1}(x)} \tag{2.37}$$

By choosing a function increment most correlated with $-g_t(x)$, the researcher replaces a potentially difficult optimization task with a classic least-squares minimization problem:

$$\theta_t = \arg\min_{\theta,\beta} \sum_{i=1}^{N} [-g_t(x_i) + \beta h(x_i, \theta)]^2 \tag{2.38}$$

Once the new weak learner is identified, the optimal step size ρ_t is calculated to minimize the loss along the direction of the gradient:

$$\rho_t = \arg\min_{\rho} \sum_{i=1}^{N} \Psi(y_i, \hat{f}_{t-1}(x_i) + \rho h(x_i, \theta_t)) \tag{2.39}$$

The model is then updated, and the process is repeated for a specified number of iterations [33]. This methodology shares historical connections with cascade correlation neural networks, which can be viewed as a specialized type of GBM

in which the base learner is a single neuron, and the loss function is the standard squared error. Ultimately, the exact form of the gradient boosting algorithm remains highly flexible, depending entirely on the chosen design of the loss function $\Psi(y, f)$ and the base-learner model $h(x, \theta)$ [33].

2.5.4 Neural Networks

In the domain of clinical informatics, ANNs operate as highly parameterized, non-linear function approximators capable of modeling complex latent interactions within high-dimensional patient data [29]. Unlike parametric or semi-parametric models, which require explicit specification of interaction terms, deep feed-forward networks inherently construct hierarchical feature representations through successive layers of weighted connections and non-linear activation functions [29].

Adapting ANNs for survival analysis requires modifying standard architectures to accommodate right-censored data. Early methodologies, such as the Faraggi-Simon network [34], proposed replacing the linear predictor $\beta^T x$ of the Cox Proportional Hazards model with the output of a neural network. These initial architectures frequently failed to outperform linear Cox models due to the absence of modern regularization techniques and the vanishing gradient problem.

Contemporary architectures, notably DeepSurv [35], address these historical limitations. DeepSurv functions as a deep feed-forward neural network that predicts the effects of patient covariates on the hazard rate, parameterized by the network's weights, θ. The architecture comprises an input layer for baseline clinical data x, followed by fully connected hidden layers utilizing modern activation functions (e.g., Rectified Linear Units) and dropout layers to mitigate overfitting.

The terminal layer consists of a single node with a linear activation function that estimates the log-risk function $h_\theta(x)$. To handle censored survival data, the network optimizes an objective function derived from the Cox partial likelihood. DeepSurv is trained by minimizing the average negative log partial likelihood, incorporating L_2 regularization to penalize large weights:

$$l(\theta) = -\frac{1}{N_{E=1}} \sum_{i:E_i=1} \left(h_\theta(x_i) - \log \sum_{j \in \mathcal{R}(T_i)} e^{h_\theta(x_j)} \right) + \lambda ||\theta||_2^2 \tag{2.40}$$

By optimizing this loss function, the network models high-level interactions between theoretical covariates—such as oncological staging, histological gradings, and treatment modalities—without requiring prior domain-specific feature engineering [35]. Consequently, models like DeepSurv function as personalized treatment recommender systems. By calculating a recommender function $rec_{ij}(x) = h_i(x) - h_j(x)$, the architecture isolates which specific therapeutic intervention yields the maximum reduction in mortality risk for an individual patient profile.

2.6 Evaluation Metrics

The validation of predictive models in survival analysis requires a statistical framework to account for the constraints imposed by right-censoring. This section provides a theoretical overview of the metrics used to assess prognostic performance across three primary domains. First, we examine the concept of discrimination using the concordance index to assess the ability to correctly rank individual risks. Second, the principles of calibration are defined to evaluate the agreement between predicted probabilities and observed event frequencies. Finally, the Integrated Brier Score is introduced as a measure of overall prediction inaccuracy, providing a cumulative assessment of model fit over the study horizon.

2.6.1 Concordance Index

The C-index [36] is the standard metric for evaluating prognostic models in survival analysis. Analogous to the AUROC curve, the C-index characterizes a model's discriminative ability by accounting for both event occurrence and timing [37].

The metric addresses the challenge of "censored" observations, where the event status is unknown after the observation time T_i. To distinguish between event and censoring, a binary variable Δ_i is introduced, where $\Delta_i = 1$ signifies an event and $\Delta_i = 0$ signifies censoring [37]. Harrell's estimator calculates the ratio of concordant pairs to comparable pairs within the dataset. A pair of subjects (i, j) is considered "comparable" if the order of their events can be determined. A comparable pair is "concordant" if the subject who experienced the earlier event ($T_i < T_j$) was assigned a higher risk score ($\eta_i > \eta_j$) [37]. The estimator is defined as:

$$C = \frac{\sum_{i=1}^{N} \Delta_i \sum_{j=i+1}^{N} \left[I(T_i < T_j) + (1 - \Delta_j) I(T_i = T_j) \right] \left[I(\eta_i > \eta_j) + \frac{1}{2} I(\eta_i = \eta_j) \right]}{\sum_{i=1}^{N} \Delta_i \sum_{j=i+1}^{N} \left[I(T_i < T_j) + (1 - \Delta_j) I(T_i = T_j) \right]} \tag{2.41}$$

where $I(\cdot)$ is the indicator function [36]. A value of $C = 1.0$ represents perfect discrimination, while $C = 0.5$ indicates a performance equivalent to random chance [37].

2.6.2 Calibration

Calibration characterizes the degree of agreement between the predicted risk of an event and the actual observed frequency. Following the hierarchical framework established by Van Calster et al. (2019) [38], calibration can be assessed through four increasingly stringent levels: mean, weak, moderate, and strong.

The most fundamental level, mean calibration (or calibration-in-the-large), involves comparing the average predicted risk with the overall observed event rate.

General overestimation is identified when the average predicted risk exceeds the event rate, while underestimation occurs when the observed rate is higher than the predicted average [38].

Weak calibration is evaluated through the calibration intercept and the calibration slope. The intercept has a target value of 0; negative values indicate general overestimation, while positive values indicate underestimation. The calibration slope evaluates the spread of the estimated risks with a target value of 1. A slope < 1 suggests that the estimated risks are too extreme (too high for high-risk subjects and too low for low-risk subjects), whereas a slope > 1 indicates that the risk estimates are too moderate [38].

Moderate calibration implies that estimated risks correspond directly to observed proportions. This level is visually assessed using a flexible calibration curve that illustrates the relationship between the estimated risk (x-axis) and the observed proportion of events (y-axis). A curve positioned near the 45-degree diagonal indicates that predicted risks correspond well to observed proportions. To obtain a precise calibration curve, a sufficiently large sample size is required, with recommendations suggesting a minimum of 200 subjects with the event and 200 without. In smaller datasets, it is considered defendable to evaluate only weak calibration by calculating the intercept and slope [38].

Finally, strong calibration represents the utopic goal where the predicted risk corresponds to the observed proportion for every possible combination of predictor values. Although this implies perfect calibration, it is rarely achievable in clinical research [38].

Notably, the use of the Hosmer-Lemeshow test for assessing calibration is increasingly discouraged. The test possesses several drawbacks, including the artificial grouping of subjects into risk strata, low statistical power, and the provision of P values that are uninformative regarding the specific nature and extent of miscalibration [38].

2.6.3 Integrated Brier Score

The Brier score provides a measure of overall prediction inaccuracy by comparing estimated patient-specific survival probabilities with the observed outcome. For a fixed time point t^*, the empirical Brier score represents the mean square error of prediction between the observed survival status $I(T > t^*)$ and the estimated event-free probability $\hat{\pi}(t^*|\tilde{X})$ [39].

To account for right-censored data, Graf et al. [39] utilize the weighted Brier score (BS^c). This approach employs IPCW to compensate for information loss, using the Kaplan-Meier estimate of the censoring distribution, $\hat{G}(t)$. The weighted score

at time t^* is formally defined as:

$$BS^c(t^*) = \frac{1}{n}\sum_{i=1}^{n}\left[(0-\hat{\pi}(t^*|\tilde{X}_i))^2\frac{I(T_i \le t^*, \Delta_i = 1)}{\hat{G}(T_i)} + (1-\hat{\pi}(t^*|\tilde{X}_i))^2\frac{I(T_i > t^*)}{\hat{G}(t^*)}\right] \tag{2.42}$$

where T_i is the observed time, Δ_i is the event indicator, and $\hat{\pi}$ denotes the predicted event-free probabilities. Following the categorization logic in [39], observations where the event occurred before t^* are weighted by $1/\hat{G}(T_i)$, while those known to be event-free at t^* are weighted by $1/\hat{G}(t^*)$. Individuals censored before t^* contribute a weight of zero as their status at t^* is unknown [39].

To assess performance across the entire study horizon, the IBS is calculated by integrating $BS^c(t)$ with respect to a weight function $W(t)$:

$$IBS = \int_0^{t_{max}} BS^c(t)dW(t) \tag{2.43}$$

The IBS ranges from 0 to 1, with 0 representing perfect prediction. A trivial, constant prediction of 0.5 results in an IBS of 0.25, which serves as a benchmark for non-informative models. Furthermore, the IBS can be used to derive an R^2-type measure of explained residual variation, $R^2 = 1 - BS^c(t^*)/BS_0^c(t^*)$, where BS_0^c is the Brier score of a null model using the pooled Kaplan-Meier estimate as a constant prediction for all patients [39].

2.6.4 .632 Estimator

The .632 estimator is a nonparametric method designed to improve the estimation of the true error rate (Err) by correcting the optimism bias inherent in the apparent error rate ($\bar{err}$) [40].

The estimator is defined as a weighted average of the apparent error rate and the leave-one-out bootstrap error rate:

$$\hat{Err}^{(.632)} = 0.368 \cdot \bar{err} + 0.632 \cdot \hat{\epsilon}^{(0)} \tag{2.44}$$

The component $\hat{\epsilon}^{(0)}$ represents the bootstrap expected error rate calculated specifically for training cases x_i that do not appear in a given bootstrap sample (i.e., cases where the bootstrap weight $P_i^* = 0$) [40]. The specific weighting of 0.632 is derived from the probability that a particular observation is included in a bootstrap sample of size n, which is $1-(1-1/n)^n \approx 1-e^{-1} \approx 0.632$ [40].

From a heuristic perspective, the estimator balances the "zero-distance" error rate ($\bar{err}$) with the error rate of points that are, on average, too far from the training set ($\hat{\epsilon}^{(0)}$), thereby approximating the expected distance of a future observation [40]. In comparative sampling experiments with small datasets ($n = 14$ to 20), the .632

estimator has demonstrated superior performance over both cross-validation and the standard bootstrap methods [40]. This efficiency is primarily attributed to a significant reduction in MSE, facilitated by the estimator's lack of negative correlation with the optimism term. Despite these empirical advantages, the theoretical justification for the .632 weighting remains less robust than that of other resampling frameworks and is recommended for application with caution [40].

2.6.5 Schoenfeld Residuals

Schoenfeld residuals constitute a diagnostic tool specifically designed to validate the PH assumption within the Cox regression framework [41]. Unlike standard residuals in linear modeling, a partial residual is defined for each i at their specific failure time t_i [41].

The residual vector $\hat{r}_i = (\hat{r}_{i1}, \ldots, \hat{r}_{ip})'$ represents the difference between the observed covariate values for the individual who failed and the conditional expectation of those covariates across the risk set R_i [41]. Formally, the kth component of the residual for individual i is calculated as:

$$\hat{r}_{ik} = X_{ik} - \hat{E}(X_{ik}|R_i) \tag{2.45}$$

where $\hat{E}(X_{ik}|R_i)$ is the weighted average of the kth covariate over all individuals under observation at time t_i, with weights determined by the estimated coefficients $\hat{\beta}$ [41].

A primary advantage of this formulation is that the residuals do not involve the estimated baseline hazard function, which simplifies their asymptotic distribution and ensures they are asymptotically uncorrelated [41]. Diagnostics are performed by plotting $\hat{r}_{ik}$ against time t_i [41]. If the proportionality assumption holds, the residuals should be centered around zero with no discernible temporal trend [41]. Conversely, systematic deviations or patterns in the plot suggest that the covariate's effect β_k varies over time, thereby signaling a violation of the PH assumption [41].

Analysis of Czech Salivary Gland Database

3

This chapter presents a technical and clinical audit of the CSGDB. The evaluation begins by exploring the application's architecture through the first three levels of the C4 model [42]: Context, Container, and Component. Following the architectural review, the analysis traces the data lifecycle, mapping the workflow from the oncologist's initial clinical entry to persistent storage.

Central to this chapter is an examination of the legacy database schema, identifying critical structural pitfalls while profiling the quality and "modeling potential" of the stored clinical variables. Furthermore, the chapter assesses the system from a legal and regulatory perspective, evaluating how data confidentiality and security are defined within the Czech Republic's healthcare framework and how these requirements are implemented in the current system.

The chapter concludes with a synthesis of the identified architectural and data-related gaps, establishing the technical requirements and motivation for the redesign and machine learning integration presented in Chapter 4.

3.1 System Architecture Audit

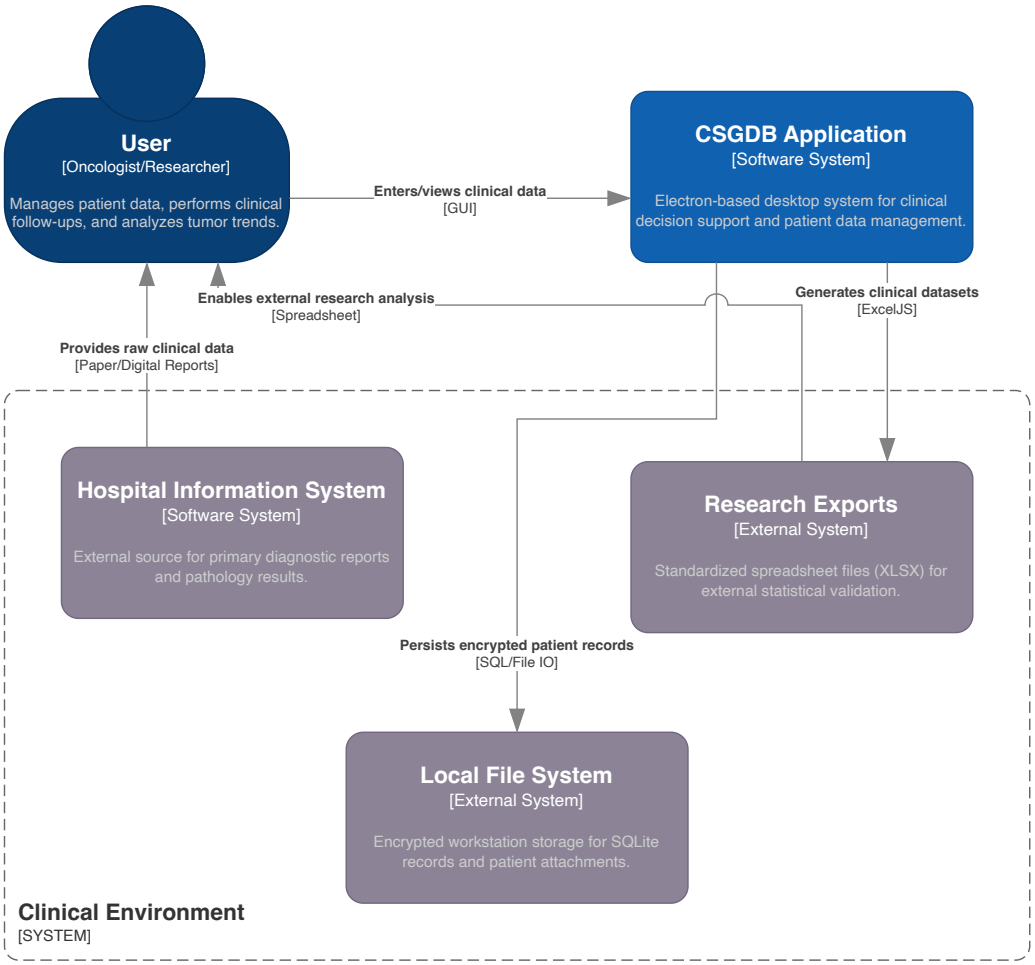


Figure 3.1: Level 1 (context) architecture of CSGDB.

At the highest level, the CSGDB works within a complex clinical environment, as shown in the figure 3.1. This setup involves three main external entities that interact with the application to support both daily patient care and long-term research. The Medical Professional acts as the central figure in this process, managing how data flows into and out of the system. Usually, diagnostic information is pulled from the HIS and entered manually into the CSGDB. This manual step is important because it allows the doctor to filter out irrelevant details, ensuring the database remains focused on high-quality research data.

Once this data is entered, it is stored directly on the local filesystem using an encrypted SQLite database. Storing the data locally rather than in the cloud is a specific choice made to protect patient privacy and follow GDPR rules, keeping

sensitive tumor records under the physical control of the medical staff. This architecture also supports a recurring research cycle: clinicians can export data as `.xlsx` files to perform specialized statistical tests in other software. The results of these external tests can then be used to refine or update the records within the CSGDB, creating a steady process of improving the dataset.

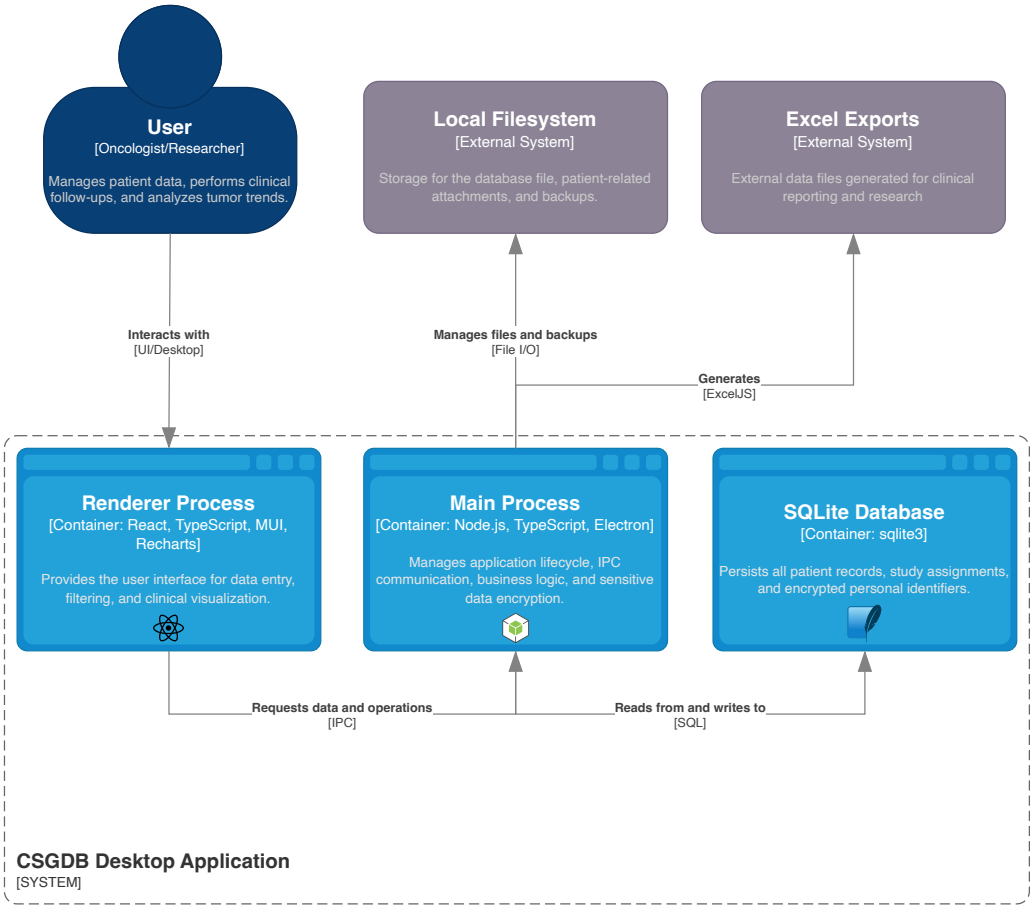


Figure 3.2: Level 2 (container) architecture of CSGDB.

Moving to the Container Diagram (Level 2) in Figure 3.2, we can see the specific technologies that form the application's foundation. The CSGDB is built on the ElectronJS framework, which organizes the system into three distinct containers: the Renderer Process, the Main Process, and the SQLite Database. This architecture is dictated by the Electron security and process model, which isolates the user interface from the underlying operating system logic [43].

The Renderer Process is implemented using React and TypeScript. This choice allows for a component-based architecture that is both maintainable and highly responsive, providing the medical professional with a smooth interface for data

entry. Conversely, the Main Process is written in Node.js and manages the system-level tasks. It acts as the central hub, using the `sqlite3` library to communicate with the local database and the Node filesystem API to handle file-based attachments. Furthermore, the Main Process is responsible for generating the `.xlsx` exports used for research, ensuring that these resource-intensive file operations do not interfere with the performance of the user interface.

3.1.1 Data Workflow

A more granular analysis of the system components (see Figure 3.3) reveals how information is handled as it moves through the application. When a user enters data, it is not sent directly to the database; instead, it is marshaled across the IPC bridge. This asynchronous mechanism ensures that the user interface remains non-blocking, even during resource-heavy operations. Once the request reaches the Main Process, a specific service—such as the Patients Manager—is selected to process the data based on the nature of the request. Before being committed to the SQLite Database, sensitive clinical variables undergo a transformation within the Security Module. Here, data is encrypted to ensure that patient records are never stored in a readable, plain-text format on the local disk. The final step in the workflow involves the Database Controller, which executes the necessary SQL commands.

A key feature of this architecture is that the database model is entirely driven by schema constants. By centralizing the definition of tables and columns into a single source of truth, the application ensures strict data integrity across both the frontend and the backend. This metadata-driven approach makes the system highly maintainable; for instance, adding new clinical variables for future predictive models only requires updating the constants rather than rewriting the core database logic.

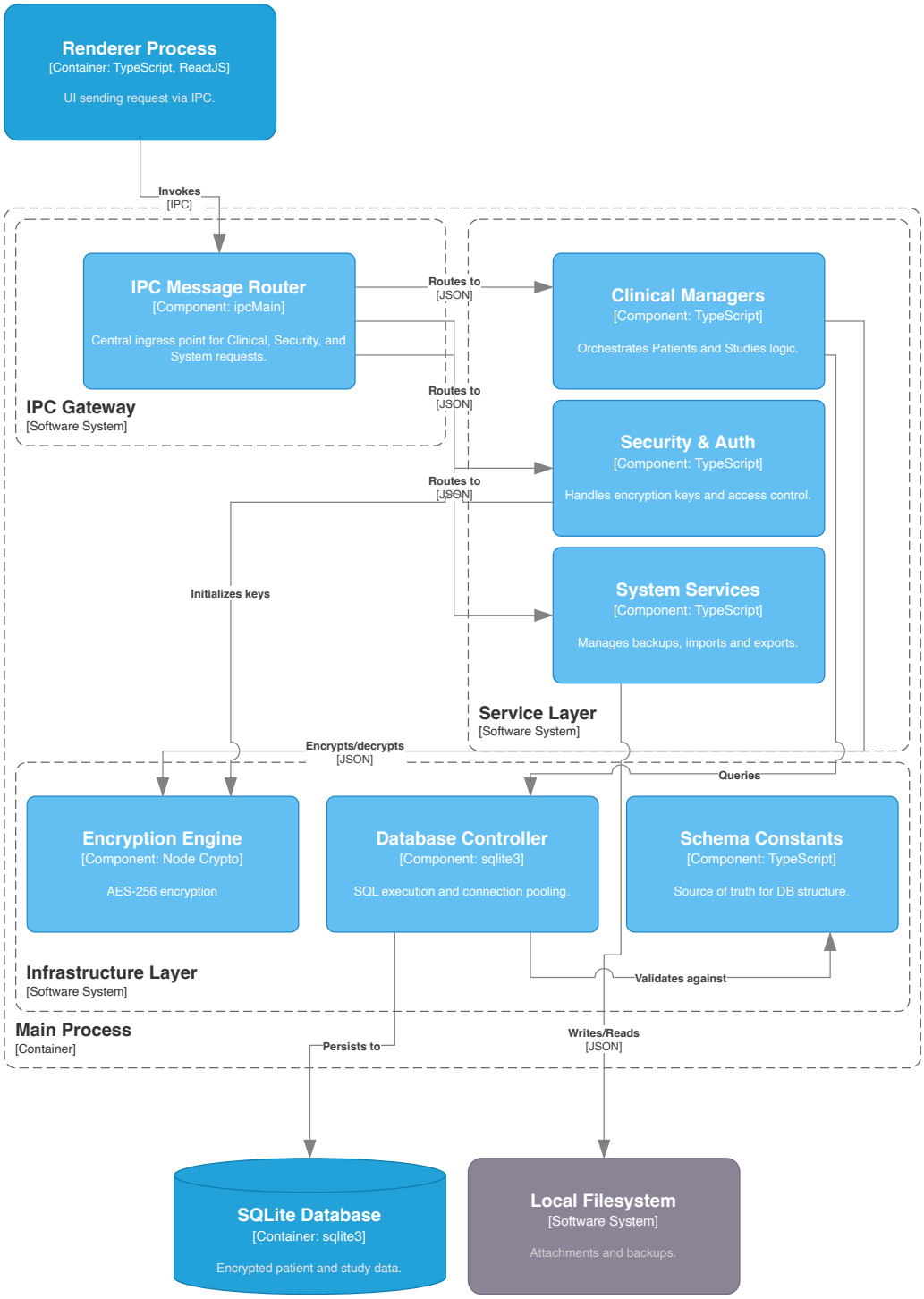


Figure 3.3: Level 3 (component) architecture of CSGDB.

3.1.2 Database Schema

The persistence layer of the CSGDB is composed of eight tables, each serving a distinct role in data management (see Figure 3.4). The core of the application revolves around three tables dedicated to malignant salivary gland tumors, which were later supplemented by tables for benign cases. These tables function as flat-file records, each representing the clinical and demographic profile of a single patient encounter. Patient records are organized into research cohorts through the `is_in_study` table, while application-level settings (such as password hashing and encryption toggles) are maintained in a singleton `password` table.

A critical observation of this schema is the extensive duplication of columns across all tumor-related tables (the complete attributes are detailed in Appendix A). This horizontal redundancy is a legacy of the system's initial development, where independent forms were created for simplicity without a centralized patient entity. While this approach allowed for rapid early deployment, it presents a significant architectural limitation for future scalability. Maintaining data integrity across these disconnected tables is difficult, and this "siloed" structure complicates the unified data extraction needed for advanced prognostic modeling.

From a technical implementation standpoint, the database utilizes SQLite, which offers a lightweight, serverless persistence layer suitable for a standalone desktop application. However, in this legacy configuration, referential integrity is largely managed at the application level rather than through database-level foreign key constraints. Data types are predominantly stored as unstructured text or integers, requiring the application logic to perform complex parsing and manual validation. This reliance on the application to enforce data rules increases the risk of inconsistencies—such as mismatched date formats or varied categorical strings—which necessitates a rigorous data cleaning phase before any statistical modeling can begin. These architectural vulnerabilities in the legacy schema served as the primary motivation for the comprehensive database redesign and normalization process presented in Chapter 4.

Figure 3.4: CSGDB simplified ERA model.

3.2 Baseline Data Profiling

Data entered into the application are of a retrospective character, meaning that for the majority of records, the full clinical trajectory—from initial diagnosis to final follow-up—is already documented. While this provides a rich foundation for training prognostic models, it also means the data was collected under varying clinical standards over several years. This section audits the existing dataset to determine

its "Modeling Potential," identifying which variables are robust enough for feature engineering and which suffer from the "sparsity" inherent in legacy medical records.

3.2.1 Clinical Variables

The CSGDB collects a complex array of oncological data, which can be categorized into six primary groups essential for longitudinal analysis:

- **Anamnesis and Demographics:** General patient information including age, sex, and residential region, alongside medical history and behavioral risk factors such as smoking habit and alcohol abuse.
- **Diagnostics:** Data regarding the initial clinical presentation, including the anatomical side of the finding and the results from various diagnostic modalities (e.g., FNAB, Core Needle Biopsy, or open biopsy).
- **Therapy Protocols:** Detailed records of the primary treatment strategy, distinguishing between surgical interventions and non-surgical modalities like adjuvant radiotherapy or chemotherapy.
- **Histopathology:** Post-operative findings that provide the "ground truth" for the tumor, including the specific histological subtype, tumor size, and status of surgical margins.
- **TNM Classification:** Standardized staging metrics (Tumor, Node, Metastasis) which serve as the baseline for most traditional prognostic assessments.
- **Follow-up and Outcomes:** Longitudinal markers tracking the patient's vital status, presence of recurrence or persistence, and the specific dates used to calculate overall and disease-free survival.

Due to the high granularity of the database, a comprehensive data dictionary containing the full list of clinical attributes for malignant patient cohorts is provided in Appendix B.

3.2.2 Quality & Sparsity

The dataset within the CSGDB is derived from retrospective HIS records, which contributes to a near 100% fill rate for core clinical attributes. Critical prognostic variables—including patient age, TNM classification, and histological subtype—are consistently populated across the malignant cohort. This high degree of data density is a significant asset, as it allows for a "complete-case" analysis and reduces the biases often introduced by data imputation methods.

Despite the high occupancy of the data fields, the dataset exhibits categorical sparsity (class imbalance). While the overall records are complete, certain tumor subtypes and anatomical locations occur much less frequently than others. This long-tail distribution of rare findings represents a "sparsity of examples" rather than "missing data," a challenge that requires specific attention during the model training and validation phases discussed in Chapter 4.

Regarding temporal integrity, the time-to-event data is complete for the majority of the entries. However, a specific characteristic of this oncological dataset is the presence of right-censored observations. For many patients, no "Event" (such as recurrence or death) has occurred within the follow-up window. This is not a failure of data quality, but a standard longitudinal reality; these censored records remain vital for the survival analysis as they contribute to the "Total Time at Risk" calculated by the models.

Finally, the data quality is bolstered by the system's input design. Most critical entries were implemented via constrained selection (e.g., checkboxes and dropdown menus) rather than free-text fields. This standardized data entry process significantly minimized human error and semantic noise, ensuring that the categorical data is ready for vectorization without the need for extensive manual cleaning or natural language processing.

3.3 Security & Legal Review

The processing of sensitive oncological data within the CSGDB is strictly governed by the GDPR [44] and the Czech Act on Personal Data Processing (No. 110/2019 Sb.) [45], specifically in its effective version including amendments up to Act No. 230/2025 Sb. Within the institutional framework of Motol University Hospital (FN Motol), the application's data handling protocols are aligned with Directive IOS 1/2011-5 ("Klinický výzkum") [46], as provided in Appendix C. In accordance with Section 4.5 of this directive, every patient record is supported by a written Informed Consent [47, 46]. This document serves as the legal manifestation of the patient's will to participate, authorizing the processing of personal identifiers—including name, date of birth, permanent address, and national identification numbers (*rodná čísla*)—exclusively for the purposes defined in the research protocol [47, 46]. The system architecture operationalizes the legal principle of purpose limitation, ensuring that sensitive "special category" data are restricted to the approved clinical study environment and compatible research objectives [47].

3.3.1 Data Confidentiality and Localized Persistence

The CSGDB maintains data confidentiality through an architectural model that isolates clinical management functions from research utility, adhering to the data management standards established by FN Motol [46]. At the workstation level, the application serves as a localized EMR for the treating physician. To facilitate longitudinal follow-up, the local SQLite database persists PII, including surnames, birth dates, and national identifiers [47, 46]. This local-only storage model is a technical enforcement of the institutional requirement that identity-revealing data "shall not leave the physician's workstation" during the course of the study [47]. To safeguard these data at rest, the database employs AES-256 symmetric encryption, satisfying institutional requirements for access control, change logging, and data integrity, as mandated for all clinical research activities within the hospital's secure environment [46, 44].

To support research and statistical analysis while adhering to the principle of Data Minimization, the application provides an export mechanism that redacts identifying information from the dataset [46]. When generating clinical exports for statistical processing, the system automatically strips identity-revealing attributes, such as patient names and national identifiers, and replaces them with the literal string "anonymized" [47]. This hard-coded redaction ensures that any data leaving the physician's workstation for research purposes is pseudonymized at the point of origin. Consequently, the exported datasets contain only the clinical variables necessary for statistical evaluation, fulfilling the institutional mandate to ensure that the participant's identity is not publicly accessible, even if results are published or shared with partners [46, 47].

3.4 Synthesis & Gap Analysis

Based on the preceding audit of the CSGDB architecture and its associated ERA model, a critical gap in the system's design becomes evident (see Figure 3.4). The current data layer exhibits a significant lack of normalization and an extensive degree of horizontal redundancy. Such structure–where clinical data is fragmented across gland-specific tables–presents a substantial technical hurdle for the data extraction and feature engineering required to train robust machine learning models.

When evaluating the system through the C4 model, the high-level separation between the "Client" and "Server" appears conceptually sound, utilizing IPC for state management [42]. However, a deeper analysis of the CSGDB source code reveals that this separation is largely superficial. In practice, there is a tight coupling between the ReactJS frontend and the Node.js main process. This structural rigidity complicates the implementation of new analytical features, as any change to the data

schema requires manual, coordinated updates across both the UI components and the backend IPC handlers.

3.4.1 Implementation Debt and Structural Coupling

Analysis of the CSGDB source code reveals several implementation issues that compromise the application architecture. These flaws contribute to a significant "tight coupling" between the ReactJS frontend and the Node.js backend, complicating further extension with machine learning techniques. When evaluated against the C4-model framework, the system's technical debt can be categorized into four primary architectural "anti-patterns":

- **Leaky Abstractions:** The legacy implementation suffers from a leaky abstraction where the frontend is directly aware of the internal SQLite schema. Clinical data identifiers are frequently hardcoded in the UI components using Czech-language strings (e.g., `jmeno`, `rodne_cislo`). This ensures that any modification to a database column necessitates manual, error-prone updates across the entire React component tree.
- **IPC Contract Fragility:** The IPC bridge lacks a formal definition. Most data transfers utilize generic `JSON` or `any` types within the `preload.ts` script. This absence of strict typing means that structural mismatches between the backend's output and the frontend's expectations are only caught at runtime, leading to cryptic crashes in the Electron Main process.
- **Positional Argument Hazards:** Communication between processes relies heavily on positional arguments passed as manually constructed arrays (e.g., `[patientId, patientType, studies]`). This pattern is hazardous; an accidental swap of indices in either the frontend or backend results in silent data corruption or logic failures that are nearly impossible to trace in a clinical setting.
- **Monolithic Type Definitions:** The structural fragmentation observed in the database is replicated within the codebase. A monolithic type definition file mixes low-level database schema definitions with high-level UI state types. This creates a high risk of circular dependencies where a change in the backend logic triggers unnecessary recompilations of the entire frontend, violating the principle of Separation of Concerns.

The synthesis of these findings highlights a clear trajectory for the redesign phase. The transition from a data-entry tool to a prognostic system requires a complete decoupling. These identified gaps—specifically the lack of type safety and the rigid

IPC layer—serve as the primary requirements for the "Contract-First" architecture and normalized data layer presented in Chapter 4.

Design and Methodology of Prognostic Models

4

This chapter details the methodology and architecture for integrating predictive survival models into the CSGDB. The objective is to transition the retrospective repository into a CDSS providing patient-specific prognostic assessments for SGC.

Three constraints dictate the methodology: the scarcity of high-dimensional data, clinical interpretability requirements, and offline security protocols at Motol University Hospital. The system utilizes a dual-model approach to meet these requirements: the Cox PH model ensures transparency, while RSF captures non-linear feature interactions.

Subsequent sections detail model selection criteria, the transition to a sidecar architecture, and preprocessing pipelines for statistical stability. The chapter further establishes decentralized training logic to mitigate "cold start" issues and ensures long-term model relevance through automated retraining triggers.

4.1 Model Selection

We surveyed the six candidate paradigms derived in Chapter 2 and selected the two that best balance statistical robustness on a sparse cohort with clinical interpretability. The selection process prioritized models that minimize structural risk while maintaining alignment with established oncological prognostic standards.

We selected the Cox PH model as the foundational semi-parametric baseline. Its utility in the context of the CSGDB stems from its invariance to the baseline hazard; it permits the estimation of covariate effects without requiring an a priori assumption about the underlying survival distribution [25]. This is particularly advantageous given the histopathological diversity of salivary gland carcinomas, where a single global distribution—such as Weibull or Log-logistic—may fail to capture the heterogeneous hazard profiles of rare subtypes. Furthermore, the re-

sulting hazard ratios provide a standardized metric of relative risk that is directly compatible with existing clinical workflows and literature.

In contrast, parametric AFT models were rejected following consultation with clinicians from Motol University Hospital. While AFT models can offer higher statistical efficiency in small samples, they quantify clinical effects through time ratios. Compounding this interpretative misalignment, the statistical validity of the AFT framework remains strictly contingent upon correct distributional specification [22].

For the machine learning component, we selected RSF to capture non-linear feature interactions and address potential violations of the proportional hazards assumption. We prioritize RSF over boosting-based methods and deep learning architectures due to its inherent regularization properties [30]. The stochastic bagging mechanism of the RSF provides robust defense against overfitting, a primary failure mode in sparse data environments. Additionally, RSF facilitates explainability through permutation-based variable importance, satisfying the requirement for clinical transparency without the architectural complexity of post-hoc interpretability layers.

Finally, we excluded advanced architectures such as DeepSurv [35] as they are structurally underdetermined when the ratio of events to predictors is low [48]. The added complexity of hyperparameter optimization and the absence of production-grade, offline-compatible tooling for neural survival models provided no commensurate predictive advantage over the more stable combination of Cox PH and RSF.

4.1.1 RSF Justification

We selected RSF as the primary machine learning engine for the CSGDB due to several statistical and operational advantages. First, the rarity of SGCs poses a significant challenge for model training. While prognostic scoring systems in the literature often leverage cohorts of more than 1,000 patients [49, 50, 51, 52], the clinical reality at Motol University Hospital is constrained by a sparse national incidence, with only 151 new cases diagnosed annually across the Czech Republic [53]. This scarcity of data necessitates a robust model that is resistant to overfitting.

While NNs, such as DeepSurv [35], can theoretically surpass the predictive accuracy of RSF by capturing deeper abstractions, they are data-intensive and prone to overfitting when trained on sparse clinical datasets. Although overfitting in NNs can be mitigated with dropout layers and rigorous hyperparameter optimization [54], these techniques introduce complexity to the software architecture. Furthermore, NNs function as "black boxes," requiring additional interpretability layers, such as SHAP [55], to provide clinical utility. Implementing SHAP within a production-grade medical application adds a secondary layer of computational logic that may

impact the system's performance [56].

In contrast, RSF is natively capable of uncovering highly complex, non-linear interrelationships within oncological data without the need for manual feature engineering [30]. It consistently performs at or above the level of competing methods in high-dimensional survival analysis while maintaining a significantly smaller computational footprint [30]. Crucially for clinical adoption, RSF provides VIMP metrics directly [30]. VIMP enables practitioners at Motol University Hospital to verify the model's logic by observing how specific features contribute to a patient's risk score, thereby fostering clinical trust.

4.1.2 Cox PH Baseline

To provide a clinically grounded benchmark for the RSF model, we utilized the Cox PH model as the statistical baseline. Since its introduction, the Cox model has remained a widely used method for survival analysis in oncology due to its semi-parametric nature, which allows estimation of HRs without requiring a specific probability distribution for the baseline hazard [49, 50].

We included the Cox PH for two primary reasons. First, it ensures clinical interpretability; the model generates hazard ratios that allow practitioners to quantify the relative risk associated with specific prognostic factors. The hazard ratios enable clinicians to directly compare the findings of the CSGDB with established oncological literature. Second, the mathematical parsimony of the Cox model is highly compatible with the application's sidecar architecture. Once trained, the model is represented as a concise vector of coefficients, enabling near-instantaneous risk calculation in the React frontend via the Python inference engine.

However, the Cox model's reliance on the proportional hazards assumption poses a limitation in heterogeneous tumors [30]. By deploying the Cox model alongside the non-parametric RSF, the system provides a dual-perspective analysis: the Cox model offers a frequentist, log-linear interpretation of risk, while the RSF captures potential violations of proportionality and complex feature interactions that a standard regression framework might overlook.

4.1.3 Explainability

The primary mechanism for global explainability is the use of VIMP rankings derived from the RSF model. VIMP is calculated by measuring the drop in prediction accuracy when a specific feature's data is randomly permuted [30]. Features that result in the highest loss of accuracy are identified as the primary drivers of the model's logic. This allows clinicians to verify that the model has not developed a reliance on spurious correlations or data artifacts, ensuring that its findings align with established TNM staging heuristics [56].

To complement global rankings, we also leverage the Cox PH model's inherent log-linear structure to provide local, patient-specific risk attribution. Unlike neural networks, which require computationally intensive additive explanation methods such as SHAP [55], the Cox model allows direct observation of how specific covariates (e.g., a patient's age or margin status) modify the baseline hazard. This approach provides the user with evidence for each prediction, highlighting which clinical factors contributed most to a patient's individual risk score.

4.1.4 Implementation Environment

The CSGDB operates on a native web stack using Electron, React, and SQLite. Consequently, implementing predictive models directly within the native JavaScript or TypeScript environment presents the path of least architectural resistance. However, the Node.js ecosystem lacks peer-reviewed, clinically validated libraries for complex survival analysis. Implementing advanced algorithms, such as RSF or the Cox PH model, from scratch introduces unacceptable risks of mathematical error. It lacks the rigorous algorithmic validation required for clinical decision support systems.

In the domains of bioinformatics and biostatistics, the R programming environment serves as the academic standard. Therneau's *survival* package [57] and the *randomForestSRC* package [58] provide extensively validated implementations of the required models. However, embedding R within a secure, offline Electron application introduces significant architectural overhead. While it is technically feasible to bundle a portable R environment for local execution, this approach requires shipping the entire runtime directory, managing local package paths, and executing processes via `Rscript`. This methodology increases the deployment footprint and complicates the lifecycle management of the predictive engine.

Conversely, Python offers a bimodal advantage: it provides statistically validated libraries for survival analysis while providing highly optimized tooling for system-level integration. Python's ecosystem includes *scikit-survival* [59] and *lifelines* [60], which provide production-ready implementations of RSF and Cox PH models, respectively. Crucially, the CSGDB's offline security constraints mandate that all patient data processing occurs locally on the workstation at Motol University Hospital. Python satisfies this requirement efficiently by allowing compilation into a single, self-contained executable binary using tools such as PyInstaller. This eliminates the need to ship a separate runtime environment, enabling a highly stable implementation of a Sidecar architectural pattern.

4.2 System Redesign

The system redesign centers on transitioning from a monolithic structure to a decoupled architecture that uses a standalone ML Engine. This technical shift necessitates a standardized inter-process communication protocol and a normalized relational schema to ensure operational stability and data integrity. The following subsections detail the analysis of this redesign, specifically addressing the modular architectural blueprint, the JSON-based data exchange contract, and the updated ERA model required to support predictive metadata.

4.2.1 Evolution of Architecture

We illustrate the system's structural evolution by comparing the legacy architecture in Figure 3.2 with the redesigned Level 2 model in Figure 4.1. From this container-level perspective, the primary modification is adding a new ML Engine to the initial architecture. Under this paradigm, the Electron Main Process is responsible for spawning the Python-based ML Engine as a child process. To ensure the application remains portable in hospital environments, the engine utilizes Python 3 and the *scikit-survival* library, packaged as a standalone executable. This encapsulation enables survival analysis without requiring a global Python installation, while maintaining system isolation and security.

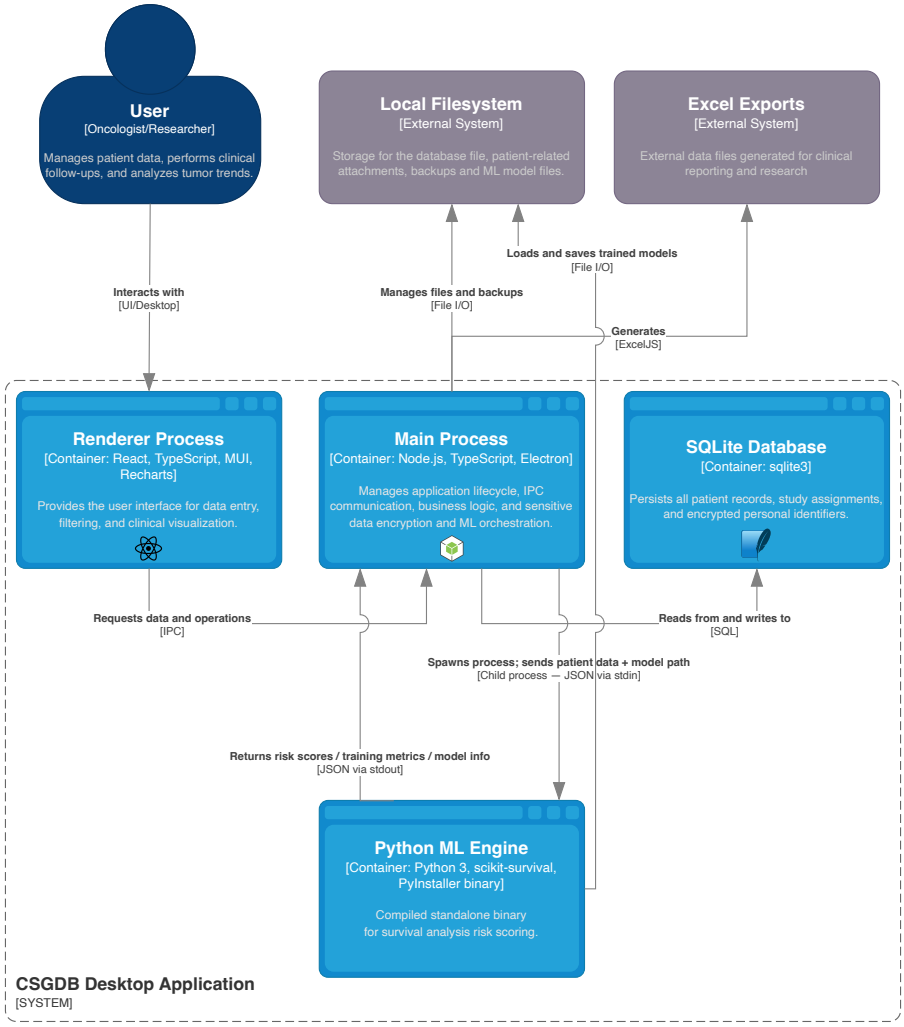


Figure 4.1: Level 2 (container) architecture of the new CSGDB application.

Examination of the Level 3 component model (Figure 4.2) reveals that the core service layer remains mostly intact, as demonstrated by comparison with Figure 3.3. The service layer was extended with a specialized ML Service that orchestrates the predictive lifecycle, leveraging the infrastructure layer—specifically the ML Reposi-

tory and ML Manager—to handle both model fitting and inference. An influential architectural refinement is the implementation of a prediction persistence mechanism; once the engine generates a prognostic estimate for a patient, the ML Service persists the output to the SQLite database. This database acts as a persistent cache, enabling near-instantaneous retrieval of results for the user interface and eliminating the need for redundant, high-latency recalculations.

Communication between the ML service and the ML engine is facilitated by the ML Manager, which transmits standardized JSON data to the Python ML Router. Upon receiving this payload, the Python engine performs internal validation and feature extraction. Furthermore, the engine can persist trained models directly to the file system, allowing selection of specific validated models and giving clinicians the option to use the most appropriate predictive model for their current patient cohort.

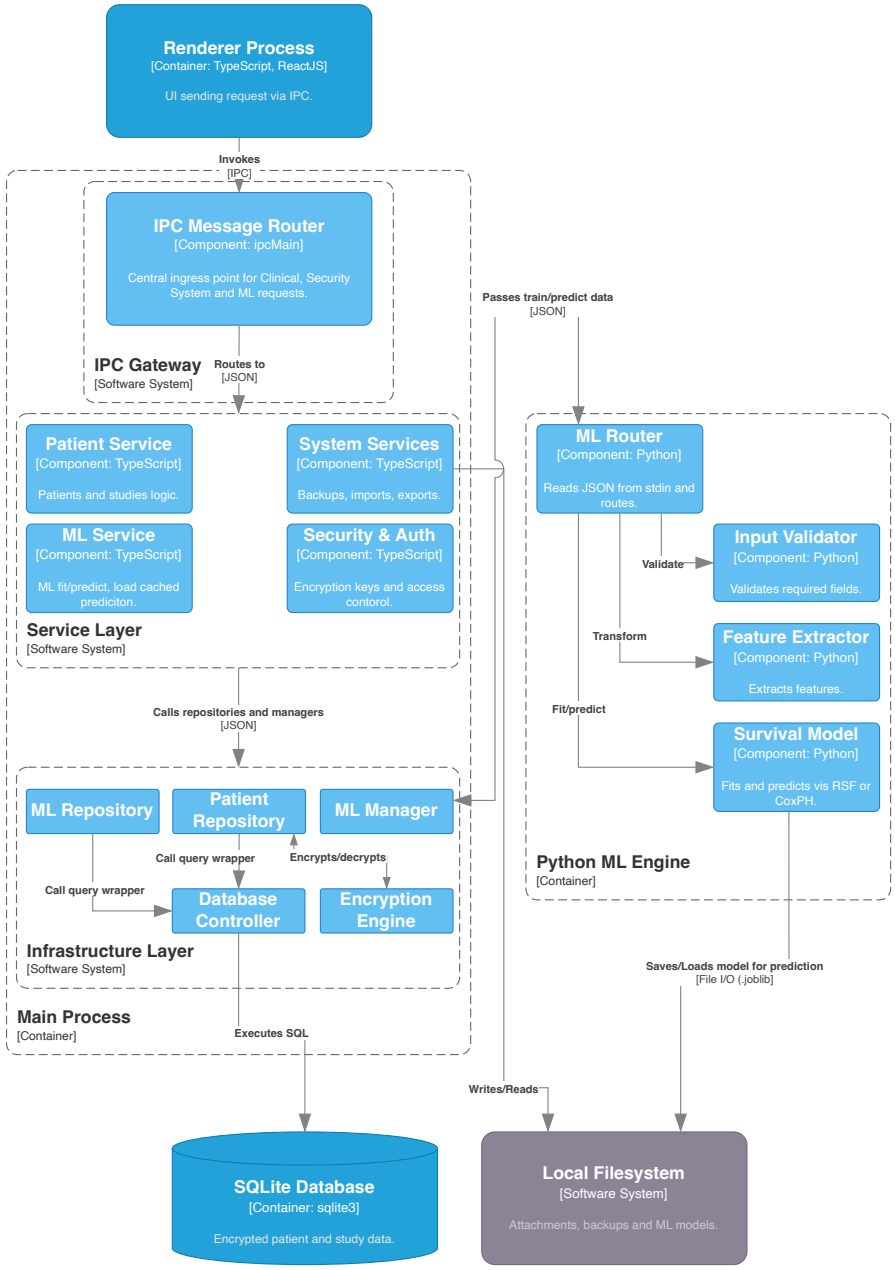


Figure 4.2: Level 3 (component) simplified architecture of the new CSGDB.

4.2.2 Data Exchange Contract

A strict data exchange contract governs the interface between the TypeScript-based backend and the Python-based ML Engine. Communication is implemented via

line-based IPC using standard input and output streams (*stdin/stdout*). In this architecture, the backend spawns the ML Engine as a subprocess via a dedicated utility, serializes clinical data into a JSON string, and writes it to the engine's *stdin*. The engine processes the request and returns a standardized JSON response to *stdout* before exiting with a status code indicating success (0) or failure (1).

The contract defines three primary operational modes, as summarized in Table 4.1. The specific behavior of the ML Engine is determined by the *mode* field within the input JSON payload.

Table 4.1: Operational modes of the ML Engine data exchange contract.

Mode	Functional Purpose
train	Fits a new survival model on a selected patient cohort.
predict	Calculates risk scores and survival probabilities for a patient.
info	Retrieves metadata and performance metrics from a saved model binary.

Input Schemas

The input schema for the *train* mode, illustrated in Listing 4.1, requires an array of patient records. This schema maps clinical variables, such as histology and TNM staging, to standardized integer IDs.

Source code 4.1: Input JSON schema for the train mode.

```
{
  "mode": "train",
  "model_type": "overall_survival",
  "algorithm": "rsf",
  "model_path": "/path/to/model.joblib",
  "data": {
    "patients": [
      {
        "age_at_diagnosis": 69,
        "clinical_t_code": "T1",
        "pathological_t_code": "T2",
        "clinical_grade_code": "Stage␣I",
        "pathological_grade_code": "Stage␣II",
        "lymphatic_invasion": "no",
        "perineural_invasion": "yes",
        "positive_node_count": 0,
        "extranodal_extension": null,
        "is_alive": true,
        "diagnosis_date": "2025-06-25",
        "death_date": null,
        "last_follow_up": "2025-12-16",
```

```
        "recidive": false,
        "date_of_first_post_treatment_follow_up": "2025-12-16",
        "date_of_recidive": null
      }
    ]
  }
}
```

The *predict* mode (Listing 4.2) uses the same field-constraint structure but targets a single patient object. This consistency ensures that the feature extraction pipeline remains uniform across both training and inference phases.

Source code 4.2: Input JSON schema for the predict mode.

```
{
  "mode": "predict",
  "model_path": "/path/to/model.joblib",
  "data": {
    "patient": {
        "age_at_diagnosis": 69,
        "clinical_t_code": "T1",
        "pathological_t_code": "T2",
        "clinical_grade_code": "Stage I",
        "pathological_grade_code": "Stage II",
        "lymphatic_invasion": "no",
        "perineural_invasion": "yes",
        "positive_node_count": 0,
        "extranodal_extension": null,
        "is_alive": true,
        "diagnosis_date": "2025-06-25",
        "death_date": null,
        "last_follow_up": "2025-12-16",
        "recidive": false,
        "date_of_first_post_treatment_follow_up": "2025-12-16",
        "date_of_recidive": null
    }
  }
}
```

Output Schemas and Clinical Interpretability

A successful execution returns a JSON response with a *success: true* flag. For the *train* mode, the result includes the C-index and a list of feature names generated after one-hot encoding. In the *predict* mode, the engine returns high-dimensional results, including normalized risk scores and survival probabilities at 1-, 3-, and 5-year intervals. As shown in Listing 4.3, the response also includes an explainability array identifying the top three risk factors based on feature importance.

Source code 4.3: Output JSON schema for overall survival prediction.

```
{
  "success": true,
  "mode": "predict",
  "result": {
    "risk_score": 0.48,
    "top_risk_factors": [
      { "feature": "age_at_diagnosis", "importance": 0.18 }
    ],
    "survival_probability_1year": 0.92,
    "survival_probability_5year": 0.52
  }
}
```

For recurrence models, the output is expanded to include both recurrence and recurrence-free probabilities (Listing 4.4), providing a more comprehensive view of the patient's prognostic outlook.

Source code 4.4: Output JSON schema for recurrence prediction.

```
{
  "success": true,
  "mode": "predict",
  "result": {
    "risk_score": 0.35,
    "recurrence_probability_5year": 0.65,
    "recurrence_free_probability_5year": 0.35,
    "top_risk_factors": [
      { "feature": "age_at_diagnosis", "importance": 0.24 }
    ]
  }
}
```

Error handling is managed through a standardized error response (Listing 4.5). If validation fails or a model file is missing, the engine exits with code 1 and returns a descriptive string, enabling the frontend to display precise feedback to the clinician.

Source code 4.5: Standardized error response schema.

```
{
  "success": false,
  "error": "Insufficient␣training␣data:␣45␣patients␣(need␣at␣least␣50)"
}
```

4.2.3 Proposed ERA Model

The redesigned ERA model, illustrated in Figure 4.3, represents a transition from a redundant schema to a normalized relational structure. A primary objective of this redesign was the elimination of duplicated tables previously used to segregate patient records by malignancy and anatomical location (e.g., *parotid_malignant* or *submandibular_benign*). These have been replaced by a table inheritance pattern centered on a unified *patient*. Specialized attributes are now managed through related tables for *benign_patient* and *malignant_patient*, reducing code duplication and ensuring consistent data handling across clinical cohorts.

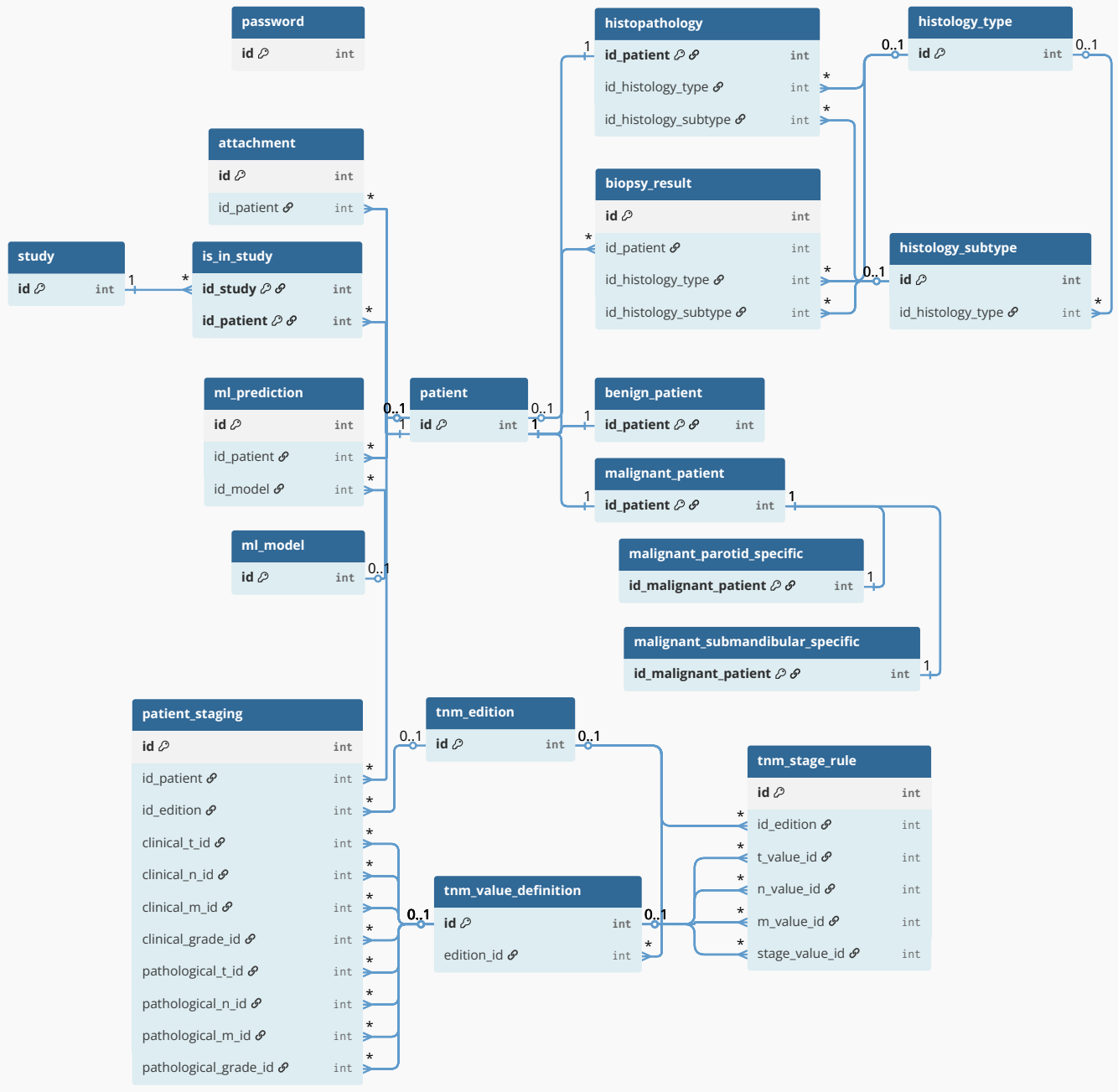


Figure 4.3: New simplified ERA model demonstrating normalized patient inheritance and ML integration.

Histology types and subtypes are now defined in independent reference tables, ensuring that both histopathology and biopsy results utilize a standardized vocabulary. Furthermore, a consolidated *attachment* table was introduced to simplify the management of binary data associated with patient records.

A critical structural extension is the introduction of the *patient_staging* table. By decoupling TNM classification from the core patient record into specialized tables—*tnm_edition*, *tnm_value_definition*, and *tnm_stage_rule*—the system gains the ability to support multiple versions of TNM. This flexibility is essential for oncology informatics, as it allows the database to incorporate updated TNM classifications without compromising the integrity of historical data. Consequently, patients followed for long durations can be assessed according to the specific classification in effect at the time of their latest recurrence or follow-up.

Finally, the schema incorporates new tables dedicated to machine learning lifecycle management. The *ml_model* table tracks trained models, including their algorithm type, training timestamp, and accuracy metrics. Complementing this, the *ml_prediction* table functions as a persistent cache for calculated prognostic estimates. This storage allows the system to maintain an audit trail of which model version generated a specific prediction and enables the application to notify clinicians when a model update necessitates recalculating a patient's risk profile.

4.3 Prognostic Variable Selection

We selected prognostic variables based on established clinical literature, data from the Czech Salivary Gland Database, and expert consensus from clinicians at Motol University Hospital. This methodology ensures that the models are medically interpretable while utilizing features with the highest independent prognostic value. Clinical consultation highlighted four indicators of particular importance— lymphatic invasion, perineural invasion, positive lymph node count, and extranodal extension—which the final seven-feature set incorporates alongside age and pathological staging. A key methodological refinement is the transition from categorical N-classification to the raw count of positive lymph nodes. This shift addresses periodic revisions to TNM standards (e.g., the transition from the 8th to the 9th edition), thereby maintaining data uniformity and predictive stability across the longitudinal dataset.

Furthermore, the limited size of the available dataset—comprising fewer than 200 patients—precludes the inclusion of all available clinical features. Utilizing an exhaustive feature set on a cohort of this scale would introduce significant noise and risk spurious correlations, thereby compromising the model's generalizability and predictive reliability. Consequently, the final feature space was restricted to the following seven variables:

- **Numeric Predictors:** Age at diagnosis and the absolute positive lymph node count are treated as continuous variables.

- **Binary Indicators:** Lymphatic invasion, perineural invasion, and extranodal extension are represented as binary integers (0/1).
- **Ordinal Predictors:** Pathological T-stage and histological grade are mapped to an ordinal integer scale ($T_1 \rightarrow 1, \ldots, T_4 \rightarrow 4$ and Stage I $\rightarrow 1, \ldots,$ Stage IVC $\rightarrow 6$), preserving the natural progression of disease severity while keeping the feature count compact. To ensure high data completeness, these features utilize a fallback mechanism where clinical staging or grading is substituted if pathological data is missing.

4.3.1 Clinical Consensus

The selection of candidate variables for the prognostic models is predicated on a consensus derived from established oncological literature. These factors are categorized into demographic, clinical, pathological, and treatment-related predictors that have demonstrated statistical significance in previous survival analyses.

Demographic and clinical indicators serve as baseline predictors in most pretreatment models. Higher age at diagnosis is consistently identified as a significant independent risk factor for recurrence and poor survival [49, 51, 52, 50]. Clinical presentation also provides early prognostic value; facial nerve dysfunction or paralysis and the presence of pain are established negative indicators, although pain is frequently associated with underlying perineural invasion [49, 51, 50]. Furthermore, visible skin invasion [49, 51, 52] and lifestyle factors—specifically smoking [52]—have been validated as independent risk factors in multivariable analyses.

Anatomical staging remains the core of prognostic assessment. Advanced T-classification (T3/T4) or tumors larger than 4 cm are strong predictors of poor outcomes in both univariate and multivariate contexts [49, 51, 61, 52, 50]. Similarly, the N-classification and the specific count of positive lymph nodes are critical independent predictors for recurrence [49, 52, 50, 61, 51]. These are specifically identified as highly significant independent predictors in multivariate analyses [49, 50, 61]. For overall survival, the presence of distant metastasis (M-stage) is a top variable significantly influencing outcomes [52].

Histopathological characteristics provide granular risk stratification. PNI is recognized as a strong independent risk factor and predictor of shorter disease-free survival [49, 51, 61, 52, 50]. In contrast, LVI often loses its independent significance in multivariate analysis [49, 50], although it remains a variable utilized in validated nomograms [51]. ENE was similarly found to lose independent significance due to heavy covariance with N-classification [49], despite its recent incorporation into the 8th edition of the UICC TNM system [51]. Additional factors such as capsular invasion [52], high-grade histology, and specific subtypes like adenoid cystic carci-

noma are associated with worse recurrence-free survival compared to low-grade tumors [61, 52, 50].

Finally, treatment-related factors significantly influence the hazard ratio. Incomplete surgical resections or positive margins drive local and distant recurrence [49, 51, 50]. Definitive surgical intervention, augmented by adjuvant radiotherapy or chemotherapy for high-risk conditions significantly improves local control and disease-free survival compared to no treatment [61, 52].

4.4 Preprocessing

The preprocessing is designed as a pipeline that transforms heterogeneous patient records into a feature matrix $X \in \mathbb{R}^{n \times 7}$. In the context of the CSGDB, the pipeline's primary objective is to bridge the gap between clinical string-based terminology (e.g., TNM codes) and the numerical requirements of survival engines.

To ensure clinical and statistical reproducibility, the preprocessing state is encapsulated within a specialized feature extractor. We serialize the state alongside the model in a `.joblib` format. This architectural decision ensures that the transformation logic applied during training is identical to that used during real-time inference in the clinical setting.

4.4.1 Feature Engineering

The feature engineering strategy utilizes a compact feature space of seven predictors. We selected this dimensionality to ensure model stability given the constraints of local hospital datasets, which often feature limited sample sizes and low event counts.

Hierarchical Fallback and Ordinal Mapping

The system implements a hierarchical fallback protocol for T-stage and the Overall Staging Group (referred to as `grade` within the implementation). Pathological data ($pTNM$) is prioritized due to its superior diagnostic precision; however, clinical data ($cTNM$) is utilized as a surrogate to prevent the systematic exclusion of non-surgical or palliative patients, thus mitigating selection bias. The derived features are defined as:

$$f_{\text{stage}} = \begin{cases} \mathcal{M}(p_{\text{code}}) & \text{if } p_{\text{code}} \neq \emptyset \\ \mathcal{M}(c_{\text{code}}) & \text{else if } c_{\text{code}} \neq \emptyset \\ \text{NaN} & \text{otherwise} \end{cases} \tag{4.1}$$

where $\mathcal{M}$ represents the ordinal mapping function. Unlike traditional one-hot encoding, these features are mapped to an ordinal integer scale (e.g., $T_1 \rightarrow 1, \ldots, T_4 \rightarrow 4$ and Stage I $\rightarrow 1, \ldots, StageIVC \rightarrow 6$). Analytically, this provides two distinct advantages:

- **Model Parsimony:** By treating staging as a single ordinal feature rather than multiple binary dummy variables, the model preserves degrees of freedom—a critical factor for the stability of models trained on small cohorts.
- **Biological Monotonicity:** This approach allows the algorithms to leverage the natural progression of the disease, assuming that an increase in stage rank correlates with a monotonic increase in cumulative hazard.

Differentiated Imputation and Scaling

The pipeline employs a single-pass median imputation strategy for all features. The median was selected as the measure of central tendency because it is robust to outliers. Following imputation, all seven features undergo Z-score standardization:

$$z = \frac{x - \mu}{\sigma} \tag{4.2}$$

While standardization is mathematically redundant for RSF, it is a strict requirement for the regularized Cox PH model. Since L_2 regularization (Ridge) penalizes the magnitude of the model coefficients, all predictors (numeric, binary, and ordinal) must be on a unit-variance scale to ensure the penalty is applied uniformly across the feature space, preventing features with larger raw variances from dominating the optimization process.

Integrity of the Survival Target

A fundamental principle of the CSGDB preprocessing logic is the non-imputation of outcome variables. Survival analysis relies on the temporal provenance of the data; specifically, the time-to-event (y_{time}) and the event indicator (y_{event}). The system explicitly excludes any patient record where a valid time origin (diagnosis or treatment date) or a terminal date (death or last follow-up) cannot be established.

Analytically, this is superior to using fixed-value fallbacks, as it avoids introducing synthetic censoring bias. For the recurrence model, if the specific treatment date is missing, the system uses the diagnosis year as a fallback temporal anchor; however, if the terminal outcome date remains unknown, the record is discarded. This ensures that the model is trained on authentic clinical observations rather than fabricated temporal data.

4.5 Training Logic

The training pipeline within the CSGDB is architected to support a decentralized deployment model, where each clinician operates a sovereign instance of the application on a local workstation. This design choice is fundamentally driven by the necessity for offline operability and strict compliance with data residency requirements, ensuring that patient records remain within the secure local SQLite persistence layer. Within this framework, we designed the training logic as an asynchronous local process where the Python engine consumes the preprocessed feature matrix to generate a specialized model artifact. To safeguard the statistical validity of these localized models, the system enforces a strict gatekeeping mechanism: the training engine remains programmatically locked until the local database reaches a threshold of $n \geq 50$ patients and at least 15 observed events. This threshold is required to ensure a baseline level of estimator stability, thereby mitigating the risk of producing unstable hazard functions, which are often characteristic of high-dimensional models trained on sparse, rare-disease datasets.

4.5.1 Hyperparameters

We fixed survival-model hyperparameters a priori to prioritize model stability and generalizability over automated local optimization. While grid search or Bayesian optimization are standard for large datasets, they risk overfitting to stochastic noise in limited cohorts like the CSGDB [62].

We configured the RSF with an ensemble of $n_{tree} = 500$ trees, comfortably exceeding the 64–128 tree convergence threshold reported by [63]. The additional margin stabilizes the permutation-based variable importance rankings, which converge more slowly than discrimination metrics, at negligible computational cost on a cohort of this size. To further mitigate the risk of the model memorizing individual patient trajectories—a documented failure mode in sparse cohorts [30]—the growth of individual trees is regularized with a maximum depth of 5 and a minimum terminal node size of 5 samples. These specific values reflect a parsimonious configuration appropriate to the limited sample size and align with the broader principle that model complexity should be bounded relative to event count in clinical prediction settings [62]. These constraints force the algorithm to identify broad prognostic clusters through the log-rank splitting rule.

In parallel, the Cox Proportional Hazards model uses L_2 regularization, motivated by the documented benefits of penalized regression for predictive models in low EPV settings [64]. The penalty strength was fixed at $\alpha = 0.1$ as a moderate, non-tuned prior; cross-validated tuning of α was avoided to prevent compounding optimism on the limited cohort. This ensures that the model coefficients remain

stable in the presence of correlated pathological predictors, preventing the extreme feature weighting that can occur in unregularized models [65].

4.5.2 Cold Start Fix

A primary challenge in this workstation-centric architecture is the lack of data on a newly deployed instance, a condition known as the cold start problem. To provide clinical utility before a local workstation meets the minimum training thresholds, the application ships with a bundled prior model. This pre-trained engine is derived from the study cohort assembled at Motol University Hospital and provides a prognostic baseline. The bundled prior serves as the active predictive tool during the database's growth phase, ensuring that clinicians have access to evidence-based risk assessments from the point of initial deployment. Once the local database surpasses the required patient and event density, the system enables a transition to a sovereign local model. In this phase, the clinician initiates a training process that operates exclusively on their own institutional data. This transition represents a shift from the Motol-derived baseline to a specialized tool that reflects the unique diagnostic nuances and surgical outcomes of the specific hospital.

4.6 Dynamic Logic

We designed the architecture of the CSGDB to facilitate a self-updating lifecycle, ensuring that the predictive engines remain temporally relevant as the local dataset matures. This dynamic logic addresses the inherent limitation of static clinical models, which often degrade in performance as institutional protocols or demographic distributions shift. By implementing a distributed update mechanism, the application enables the predictive logic to be continuously refined without requiring a full system redeployment. The system treats the trained model not as a final artifact, but as a versioned state that reflects the cumulative evidence available at a specific point in time.

4.6.1 Retraining Triggers

The retraining process is managed by automated triggers designed to balance model recency with statistical stability. To prevent stochastic fluctuations—where the model coefficients might vary excessively due to the influence of a few atypical patient records—the system avoids continuous retraining after every database modification. Instead, the application monitors the local SQLite instance for a proportional growth threshold. A retraining cycle is programmatically initiated whenever the total number of patient records increases by 10% relative to the cohort size at the time of the previous successful model training. This percentage-based trigger ensures

that the frequency of model refinement scales dynamically with the repository's volume.

To complement these automated protocols, the architecture provides a manual invocation mechanism that grants the clinician the autonomy to trigger a re-estimation cycle on demand. This feature allows the practitioner to exercise professional judgment, such as updating the model following a major retrospective data entry session or after a significant period of clinical observation that has not yet reached the 10% threshold. Whether triggered automatically or manually, the system maintains a comprehensive historical log. This audit trail allows the user to review every predictive engine from the initial bundled prior model to the latest local iteration, with each entry detailing the algorithm type (Cox PH or RSF), the target outcome (overall survival or recurrence), a timestamp, and the statistical context of the training set (n patients and e events).

Performance for each historical version is quantified through a dual-metric display of both the apparent C-index and the honest .632 bootstrap estimate. This detailed versioning allows the practitioner to verify the evidence base for past risk assessments retrospectively and to monitor the quantitative evolution of the system's predictive accuracy. Furthermore, a performance-gating mechanism remains active during both manual and automated updates, alerting the clinician if a new iteration shows a significant degradation in predictive discrimination. This ensures that the predictive engine's evolution remains under human-in-the-loop supervision, prioritizing clinical safety over automated convenience.

Integration and Deployment

5

Following the analysis, the latest version was implemented and deployed across five implementation units. First, we created the Python machine learning engine using a sidecar pattern, based on the features defined in the analysis, and implemented the RSF and Cox PH algorithms. The addition of machine learning led to a deep backend refactoring using the Repository-Mapper-Service pattern. Due to this significant backend refactor, it was necessary to update the infrastructure—primarily to remove the tight coupling between the frontend and backend and to expose risk prediction functions to the renderer process. Lastly, we developed a user interface that allows clinicians to select models and algorithms while providing clear explainability for the predictions. Additionally, we extended the frontend to support multiple TNM classification versions. Finally, we discussed the data migration options with clinicians and updated the preexisting GitHub pipelines to integrate the ML engine.

5.1 ML Engine

The integration of a machine learning engine into a clinical environment requires a high degree of algorithmic reliability. Given that prognostic outputs directly inform clinical decision-making, it is vital to utilize implementations that are both peer-reviewed and computationally stable. Consequently, developing custom survival algorithms was avoided in favor of the mature Python ecosystem. Python was selected over JavaScript (Node.js) due to its superior foundation in survival analytics and the availability of specialized libraries. Beyond its analytical capabilities, Python was also chosen for its interoperability, enabling seamless integration with the existing application via a sidecar architecture.

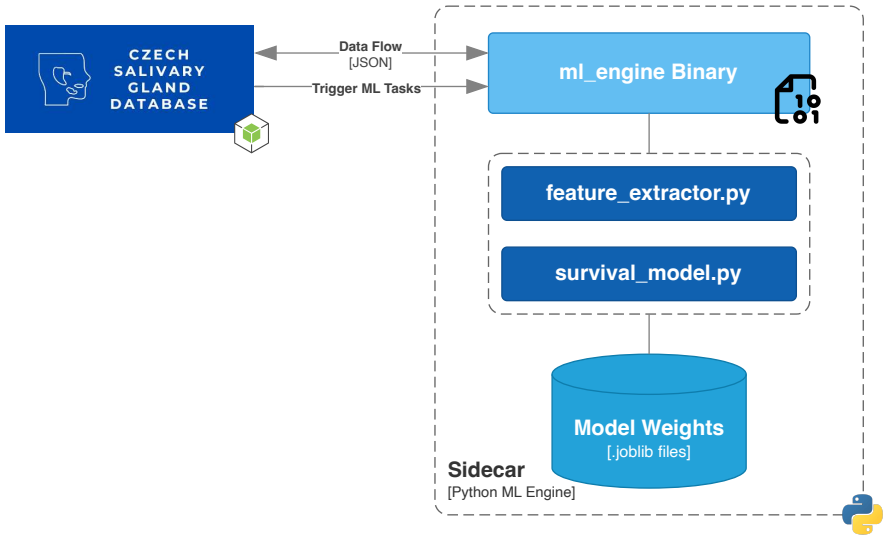


Figure 5.1: Python sidecar pattern.

5.1.1 Sidecar Pattern

The Python-based machine learning engine is integrated into the CSGDB ecosystem using a sidecar pattern (see Figure 5.1). In this architecture, the ML engine operates as a self-contained executable that runs in parallel with the main application. This approach provides a significant performance advantage: by offloading computationally intensive survival analysis to a separate process, the primary user interface remains unaffected by heavy processing loads, ensuring high responsiveness during clinical use.

The application manages the lifecycle of the sidecar via the Node.js `child_process` module. To eliminate the cold-start latency introduced by PyInstaller initialization, the main sidecar process is spawned once at application startup and maintained throughout the session. The system registers a process termination handler for the Electron `before-quit` event to ensure a clean release of resources upon exit.

To balance UI responsiveness with computational isolation, the system implements two distinct execution modes. Prognostic prediction and metadata queries are routed through the persistent sidecar process, which accepts newline-delimited JSON requests via standard input (`stdin`) and returns results via standard output (`stdout`) without restarting. Conversely, model training spawns a transient, one-shot process per job. This strict isolation prevents long-running or failed training operations from corrupting the shared sidecar state, while ensuring the PyInstaller

initialization cost is paid only once for standard predictive workflows.

Inter-process communication utilizes a split-channel design. While computational results are exchanged via standard I/O streams, real-time progress updates are emitted exclusively on standard error (`stderr`) as JSON objects containing execution state and completion percentage. The Node.js host incrementally parses these payloads and forwards them to the renderer process via a dedicated IPC push channel, effectively decoupling the progress signal from the final computational result.

Within the sidecar environment, incoming data is first handled by `feature_extractor.py`, which transforms the raw clinical variables into a format suitable for the `survival_model.py` prediction script. The model itself is loaded from the file system, where it is stored in `.joblib` format.

5.1.2 Feature Extraction

The `FeatureExtractor` class in the `python-ml-engine` module serves as the core of the data preprocessing pipeline. This component transforms raw patient entities into a structured feature space consisting of seven prognostic variables: age at diagnosis, positive node count, T-stage, overall stage, lymphatic invasion, perineural invasion, and extranodal extension. The extraction process follows a pipeline that sequentially applies ordinal mapping, binary encoding, and median imputation to ensure the survival models receive mathematically optimized data.

For continuous variables and ordinal scales, the extractor performs a synchronized numeric transformation. To maintain the integrity of the statistical distribution while accommodating missing values, the module uses median imputation via the `SimpleImputer` class. This ensures model robustness without the bias introduced by mode-based frequentist imputation. Subsequently, the pipeline applies Z-score normalization using the `StandardScaler` to numeric features. This process centers the feature mean at zero and scales the variance to one, preventing variables with larger numerical ranges, such as age, from disproportionately influencing the model's weight coefficients.

Ordinal encoding is applied to the T-stage and overall stage using a pathological-first fallback logic. The system prioritizes pathological values (pT, $pStage$); if these are absent, clinical values (cT, $cStage$) are utilized. As demonstrated in Listing 5.1, the implementation utilizes the pandas `.where()` method to enforce this priority efficiently across the entire data frame.

Source code 5.1: Pathological-first fallback logic for clinical staging.

```
patho_t = df.get('pathological_t_code', pd.Series([None] * len(df)))
clinic_t = df.get('clinical_t_code',    pd.Series([None] * len(df)))
raw_t    = patho_t.where(patho_t.notna() & (patho_t != ''), clinic_t)
```

```
df['t_stage'] = raw_t.map(self._parse_t_stage)
```

T-stage values map to an integer scale, with T1 through T4 corresponding to 1 through 4, respectively (T4a and T4b both map to 4). Overall stage maps to a six-point scale: Stage I to 1, Stage II to 2, Stage III to 3, and Stage IV sub-classifications (IVA, IVB, IVC) to 4, 5, and 6. Unrecognized codes or 'TX' designations result in `NaN` values, which are subsequently resolved by the shared median imputer.

Binary feature encoding transforms qualitative clinical observations into a numeric format suitable for the Cox and RSF engines. Clinical indicators for lymphatic invasion, perineural invasion, and extranodal extension are converted from 'yes'/'no' string literals to 1.0 and 0.0, respectively. Any missing or null values are preserved as `NaN` during the initial extraction phase, ensuring they are treated consistently with the numeric variables during the final imputation stage.

Finally, the extractor transforms clinical outcomes into two target arrays: the Event Status (E) and the Time-to-Event (T). The engine converts boolean clinical flags into binary integers, where 1 denotes an event and 0 denotes a right-censored observation. The duration (T) is calculated as the number of days between the date of diagnosis and the date of the event or last follow-up. To maintain the mathematical validity of the survival curves, the system enforces strict data integrity: patients with missing follow-up dates are excluded entirely from the training and validation cohorts, rather than assigned default durations.

5.1.3 Predictive Logic

The `SurvivalModel` class in the `python-ml-engine` module wraps the `scikit-survival` library to handle model training, risk estimation, and evaluation. This component supports two primary algorithms, configured according to the hyperparameter selection detailed in Section 4.5.1.

The RSF model uses an ensemble of 500 decision trees (`n_estimators = 500`) with a maximum depth of 5 (`max_depth = 5`). The model requires 10 samples to split an internal node (`min_samples_split = 10`) and 5 samples per leaf node (`min_samples_leaf = 5`). The `random_state` is fixed at 42 to ensure reproducibility. To reduce wall-clock training time on multi-core clinical workstations, the algorithm utilizes all available CPU threads (`n_jobs = -1`). The Cox PH model utilizes an L2 regularization penalty (`alpha = 0.1`) and Efron's method for handling tied event times. The solver is limited to 100 iterations (`n_iter = 100`).

The engine enforces a minimum threshold of 50 patients and 15 events before permitting model training. During the `fit` process, the class transforms input data into the structured arrays required by `scikit-survival`.

The engine produces a continuous Survival Function $S(t)$, which it samples at one, three, and five-year milestones. The final Risk Score is calculated as $1 - S(t =$

5 years). The concrete implementation of this temporal sampling is demonstrated in Listing 5.2.

Source code 5.2: Extraction of temporal survival milestones and risk score calculation.

```
surv_prob_1y = self._get_survival_prob(survival_funcs[0], 365)
surv_prob_3y = self._get_survival_prob(survival_funcs[0], 1095)
surv_prob_5y = self._get_survival_prob(survival_funcs[0], 1825)
risk_score   = 1.0 - surv_prob_5y
```

Performance is measured using the C-index. Additionally, the engine computes a bootstrapped C-index alongside its standard deviation, providing clinicians with a quantified uncertainty estimate for predictive discrimination. To provide transparency, the engine calculates feature importance via Permutation Importance for RSF models and the absolute value of coefficients for Cox PH models. Finally, the system uses `joblib` to serialize "model bundles" that include the trained estimator, the `FeatureExtractor` instance, and all validation metadata.

5.2 Backend Refactor

To support the integration of the machine learning engine the Node.js backend underwent a structural refactor. This transition shifted the architecture toward a modular Repository-Mapper-Service pattern. This design ensures that data persistence, business logic, and data transformation remain decoupled. By isolating these concerns, the system can manage the lifecycle of the analytical Python "sidecar" independently of the core application state.

5.2.1 Persistence Layer

The refactored persistence layer replaces direct SQL execution with a Repository Pattern, primarily through the `PatientRepository` and `MLRepository`. A specialized set of functions in `dbHelpers.ts` executes the SQL commands required by these repositories. This design choice is illustrated in Listing 5.3, where the generic type parameter <T> ensures that every repository receives compile-time type safety on query results without exposing the underlying SQLite callback API.

Source code 5.3: Generic SQL helpers for the repository layer.

```
export const runQuery    = <T>(query: string, params?: unknown[]) =>
    Promise<T | undefined>
export const runQueryAll = <T>(query: string, params?: unknown[]) =>
    Promise<T[]>
export const runInsert   = (tableName: string, data: Record<string,
    unknown>) => Promise<number>
```

To meet the application's security requirements, the `PatientRepository` integrates the `encryption.ts` utility, which encapsulates AES-256-CBC cryptographic logic. This design decouples the repository from the encryption implementation, ensuring that sensitive data is encrypted at rest while remaining accessible to the service layer in plaintext.

One notable repository is `tnmRepository`, which enables users to work with multiple versions of the TNM classification. This abstraction is critical for the application's longevity, as international oncological committees periodically update the TNM classification framework. As a final layer of abstraction, each table maps to a dedicated entity representation class, ensuring that each repository processes a strictly defined set of data types.

5.2.2 Service & Mapping

The `MLService` and the specialized mapping layer coordinate operations among the frontend, the database, and the analytical sidecar. The `MLService` implements a Result Caching and Stale State Management system to optimize performance and traceability. By storing prognostic results in an `ml_prediction` table, the service prevents redundant calls to the Python engine. Furthermore, the service tracks the specific model ID used for each prediction.

The core of this stale-detection algorithm relies on strict identifier comparison, as shown in Listing 5.4. When the UI detects a stale prediction, and the clinician confirms the need for recalculation, the service layer's `recalculate` flag bypasses the cache. It triggers a fresh inference call to the Python engine, which subsequently overwrites the stored entry with the updated prognostic score.

Source code 5.4: Stale prediction detection logic.

```
1 const cached = await mlRepository.getSavedPrediction(patientId, modelType
    );
2 result.is_stale = cached.id_model !== activeModel.id;
```

Data transformation is performed through a specific mapping pipeline that converts clinical data into standardized, machine-readable formats. While the `PatientMapper` handles standard DTO conversions, specialized mappers translate human-readable Czech clinical terms into unique integer IDs and standardized keys. Subsequently, the service utilizes a `sanitizePatientForML` method to prepare this data for the Python sidecar by aligning database strings with the specific categorical labels expected by the machine learning models. Listing 5.5 demonstrates this translation, bridging the gap between localized database fields and the standardized formats required by the Python feature extractor.

Source code 5.5: Data sanitization and mapping for the ML sidecar.

```
const sanitizePatientForML = (patient: PatientDto): MLPatient => ({
    age_at_diagnosis:    toNum(patient.vek_pri_diagnoze) ?? undefined,
    clinical_t_code:     patient.t_klasifikace_klinicka,
    pathological_t_code: patient.t_klasifikace_patologicka,
    lymphatic_invasion:  patient.lymfovaskularni_invaze_histopatologie,
    positive_node_count: toNum(patient.
    pocet_lymfatickych_uzlin_s_metastazou_histopatologie),
    is_alive:            patient.stav === 'alive',
    // ... remaining fields
});
```

Finally, the backend incorporates a functional Rule-Based Staging Engine within the `tnmRepository` that automatically calculates aggregate clinical and pathological stages (Stages I–IV) based on priority-based T, N, and M rules. As shown in Listing 5.6, this engine iterates over priority-ordered rules retrieved from the database, using a wildcard-matching design (`!rule.t_value`) to remain agnostic to the specific TNM edition being evaluated.

Source code 5.6: Priority-based TNM staging engine.

```
const rules = await getStageRulesByEdition(editionId); // ORDER BY priority
     DESC
for (const rule of rules) {
    const tMatch = !rule.t_value || rule.t_value === tCode;
    const nMatch = !rule.n_value || rule.n_value === nCode;
    const mMatch = !rule.m_value || rule.m_value === mCode;
    if (tMatch && nMatch && mMatch) return rule.stage;
}
```

5.3 Infrastructure

Integrating the machine learning engine into the Electron environment requires a specialized infrastructure to manage data flow across isolated processes. This section details the restructuring of the application's type hierarchy and the implementation of an asynchronous IPC bridge. These changes establish a stable data contract between the UI and the backend, ensuring that the system remains modular while handling the long-running execution cycles of the Python analytical core.

5.3.1 Schema Evolution

The changes to the database ER model required by the analysis in Chapter 4 led to a substantial refactoring of the IPC and the overall communication layer. The initial software used a single set of type definitions shared across the frontend, backend,

and IPC modules, which created tight coupling. To mitigate this, we applied several steps. The first, which was counterintuitive, was the deliberate duplication of frontend types into the IPC module. In this new structure, these types serve as contracts that specify exactly what the IPC module receives from the frontend and what it returns to the backend after processing.

This transition provided another functional reason to utilize the mappers mentioned in the backend refactor section. A crucial use of these mappers is transforming the newly defined IPC contract types into the database entities used by the backend.

These changes establish a clean type hierarchy and remove the direct dependency between the frontend and the backend. By providing each part of the system with its own types, the modules have become more independent and less sensitive to breaking changes during future updates to either the UI or the database schema.

5.3.2 IPC Bridge

To enable machine learning-related communication between the Renderer process and the Main process, the system utilizes a dedicated IPC module called `ipcMLHandles.ts`. This module implements the `ipcMain.handle` pattern to expose the primary functions for model training and prognostic prediction. Additionally, the bridge provides auxiliary handles to retrieve model metadata, switch to the currently active model version, and delete saved model bundles.

Because these operations are computationally intensive, the system avoids blocking the Renderer process by using asynchronous promises and a real-time push mechanism. As demonstrated in Listing 5.7, progress updates are not returned via the resolving promise, but are instead pushed directly to the frontend using `event.sender.send()`. The implementation includes a necessary `isDestroyed()` guard to prevent memory leaks or crashes if a long-running calculation outlives the user interface window.

Source code 5.7: IPC progress forwarding with window lifecycle guards.

```
ipcMain.handle(ipcMLChannels.trainModel, async (event, args) => {
    const [modelType, algorithm] = args;
    const onProgress = (progress: number, stage: string) => {
        if (!event.sender.isDestroyed())
            event.sender.send(ipcMLChannels.mlProgress, { progress, stage
     });
    };
    return await trainMLModel(modelType, algorithm, onProgress);
});
```

To prevent clinicians from being locked into lengthy computations, the IPC bridge exposes a dedicated cancellation handle (`ipcMLChannels.mlCancel`). When invoked, the backend sets a cancellation flag and terminates the child process via

`child.kill()`. The resulting promise is then rejected with a typed error (`err.canceled === true`), allowing the frontend to distinguish deliberate user cancellation from an unexpected process failure and to present appropriate UI feedback.

5.4 Clinical UI/UX

The UI/UX implementation focuses on a simple representation of the information to the clinicians. We display the risk prediction to clinicians in two components: a patient-centric component for visualizing risk predictions and a prediction configuration component. We also focused on creating an explainable UI that increases confidence in the prediction results. Finally, we made some critical changes to the TNM classification component. As in the previous version, we statically set TNM to TNM 8, and doctors have already suggested updating to TNM 9.

5.4.1 Risk Prediction

We implemented the Risk Prediction module through two primary components: `PatientRiskCard` and `MlRiskScoring`. While the `PatientRiskCard` is embedded within the patient detail view to provide clinical context, the `MlRiskScoring` component is a dedicated administrative view for managing the machine learning engine.

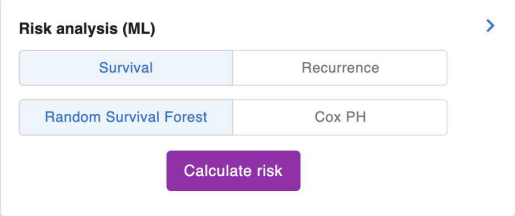


Figure 5.2: Interactive state of the `PatientRiskCard` component.

As shown in Figure 5.2, the risk card provides two primary configuration toggles. The first allows the clinician to select the prediction endpoint: Overall Survival or Recurrence-Free Survival. The second toggle switches between the available algorithms: RSF or the Cox PH model. Upon clicking the **Calculate risk** button, the component invokes the IPC bridge to trigger the Python analytical engine.

Once the calculation is complete, we render the results as a set of temporal milestones (see Figure 5.3). The UI displays the aggregate risk score alongside specific survival/recurrence probabilities at 1-, 3-, and 5-year intervals. To aid interpretability, the card also visualizes the top three most influential risk factors for that specific patient. A critical feature of this component is its version-tracking logic: the UI automatically detects whether the clinician generated the patient's prediction with an

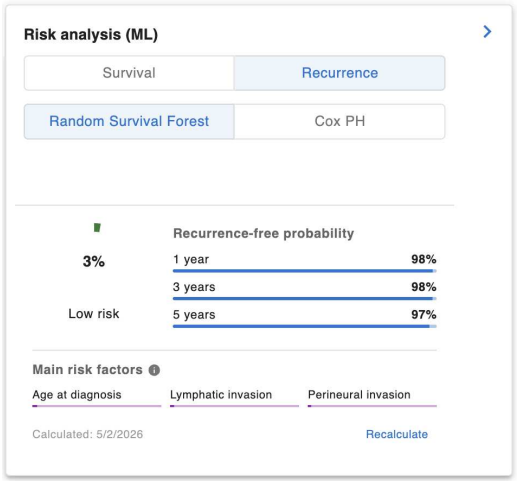


Figure 5.3: Visual output of a completed prognostic calculation.

older model version and prompts the user to recalculate the score to ensure clinical accuracy.

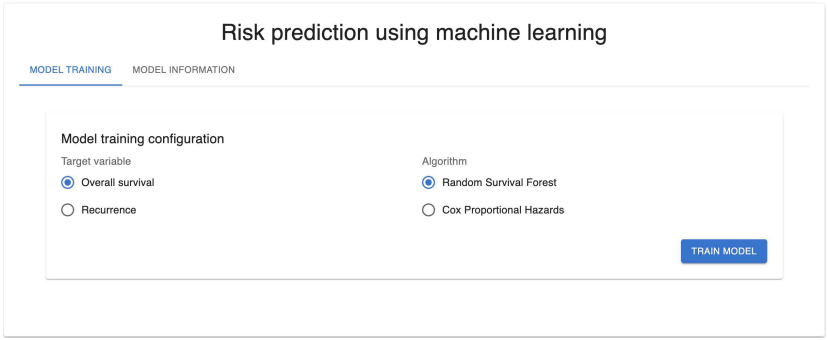


Figure 5.4: `MlRiskScoring` management interface: Model Training tab.

The administrative side of the module, `MlRiskScoring`, manages the model lifecycle. The first tab (Figure 5.4) provides the interface for training new models. After selecting a target algorithm and tumor location, the user can initiate training. Upon completion, the system renders the model's performance metrics, including the C-index and the size of the training cohort.

The second tab, labeled "Model Info" (Figure 5.5), contains the model registry. This view allows the administrator to browse the history of all trained models and strictly define which specific RSF and Cox PH bundles are currently "Active" for the entire application environment. The registry displays the algorithm type, C-index, training timestamp, and event count for each entry. Users can also delete obsolete models to manage local storage and prevent the use of outdated prognostic logic.

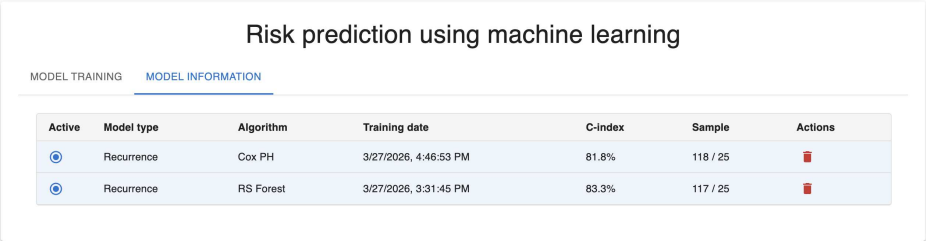


Figure 5.5: `MlRiskScoring` management interface: Model Info tab.

5.4.2 Asynchronous Operation Layer

To coordinate the asynchronous nature of machine learning operations across isolated UI components, the frontend employs a dedicated React context, `MLOperationProvider`. This context acts as the single source of truth for all ML-related process states, exposing strictly typed methods for model training, prognostic prediction, and user-initiated cancellation.

Crucially, the provider also exposes a `startTrainingQueue` function. While clinicians manually trigger single-model fits via the UI, the application's automated retraining listener uses this queue method. When a patient record is saved, the backend checks whether the current malignant cohort size has grown by 10% or more since the active models were last trained. If this threshold is exceeded, a notification is pushed to the renderer prompting the clinician to retrain. Upon confirmation, the provider's `startTrainingQueue` function sequentially re-fits all algorithm–endpoint combinations.

Whether triggered manually or automatically, the operation invokes the context method rather than interfacing with the IPC bridge directly. The context tracks execution time, parses progress updates pushed via the IPC channel, and stores a structured completion state upon resolution. This state contains the formal outcome (success, canceled, or error), the exact elapsed duration, and, in the case of individual predictions, a programmatic reference to the target patient enabling direct routing.

To maintain UI clarity during long-running tasks, an `MLProgressWidget` subscribes to this context. It renders a globally visible progress indicator alongside a cancellation control whenever an operation is active, provided no component has registered an inline display suppression. Upon completion or failure, the widget automatically transitions to a localized feedback notification before dismissing itself, ensuring clinicians are informed without requiring manual modal closure.

5.4.3 Explainability UI

The system includes an explainability layer implemented within the `PatientRiskCard` as a prioritized list of the top three clinical factors that most

significantly influence the individual patient's risk score (as shown in Figure 5.3). The layer provides "Evidence" for each prognostic result.

The system identifies these factors by extracting the variables with the highest absolute weights from the Cox PH coefficients or the greatest Permutation Importance from the RSF model. To ensure readability by clinicians, a mapping utility translates internal database keys (e.g., `p_stage_n`) and categorical values into standardized clinical terminology. This mechanism allows clinicians to verify if the model's logic aligns with medical intuition. By providing the "why" behind the percentage, the interface provides another layer of credibility.

5.4.4 TNM Classification Component

We refactored the TNM classification component to transition from a static input form to a reactive, asynchronous staging engine. Refactoring was necessary to ensure that the machine learning engine always operates on valid, up-to-date clinical stages. We split the implementation between a presentation layer (`TNMClassificationCalculator`) and a logic layer (`useTnmData` hook).

Reactive State Synchronization

The core of the component's reactivity lies in its `useEffect` chain. As shown in Listing 5.8, the component monitors the `formData` for changes to T, N, or M identifiers. When a change is detected, it triggers an asynchronous call to the `calculateStage` function provided by the custom hook.

Source code 5.8: Reactive stage calculation logic.

```
useEffect(() => {
const calculateCurrentStage = async () => {
if (!formData || isLoading) return;

    const tId = formData[tIdLabel];
    const nId = formData[nIdLabel];
    const mId = formData[mIdLabel];

    if (!tId || !nId || !mId) {
        setTnmStage(null);
        return;
    }

    const stage = await calculateStage(tId, nId, mId);
    setTnmStage(stage);
};
calculateCurrentStage();
}, [
```

```
    formData,
    tIdLabel,
    nIdLabel,
    mIdLabel,
    isLoading,
    calculateStage
]);
```

This design ensures that the derived clinical stage (e.g., Stage II) is always synchronized with the raw staging parameters. Once the stage is calculated, a second effect propagates the result back to the global `formData` state, ensuring that the persistence layer receives the correct aggregate stage without manual user intervention.

Data Orchestration via Custom Hooks

To decouple the UI from the underlying IPC infrastructure, all data fetching and staging logic is encapsulated in the `useTnmData` hook. This hook manages the complexity of the TNM versioning system discussed in Section 5.3.

As shown in Listing 5.9, the hook utilizes a `Promise.all` pattern to concurrently fetch T, N, M, and Grade definitions via the IPC bridge (`window.api`). By centralizing the staging logic—specifically the `calculateStage` callback—the hook provides a clean interface for the UI to request backend calculations without needing to understand the underlying SQL schema or the specific rules of different TNM editions.

Source code 5.9: Asynchronous data fetching in useTnmData hook.

```
const [
    tValues,
    nValues,
    mValues,
    gValues
] = await Promise.all([
window.api.getTnmValues(selectedEdition.id, 'T'),
window.api.getTnmValues(selectedEdition.id, 'N'),
window.api.getTnmValues(selectedEdition.id, 'M'),
window.api.getTnmValues(selectedEdition.id, 'G'),
]);
```

This modular approach ensures that the TNM classification remains extensible. If a new TNM edition is released, the component dynamically adapts by loading the relevant values and rules through the hook, maintaining the application's longevity and clinical accuracy.

5.5 Deployment and Adoption

Due to critical changes in the database ER model, the deployment phase is more complex than in the previous version, where users could import an entire SQLite database. Clinicians must now use a `.xlsx` export to migrate stored patient records into the system. However, even this approach required several changes to the import logic; the primary challenges arose from the new TNM classification system, which now allows the application to support multiple versions. To resolve this, we decided that imports from legacy `.xlsx` files will always assume the TNM 8 version. This assumption is clinically sound for older data, while all new exports produced by the system now explicitly include the specific version used.

One limitation of this migration is the loss of data regarding which "Studies" included which patients. In this context, studies primarily serve as patient groupings, lacking additional internal functionality. After consulting with clinicians, we determined that this was not a significant issue, as they used the feature minimally in daily practice. We settled on a practical solution: clinicians can include study information in a patient's notes before export, and then manually reassign them in the new interface. Since creating a new study with *n* patients is straightforward, this manual step does not significantly hinder the clinical workflow.

To lower the barrier to initial clinical adoption, the application is distributed with a set of pre-trained model bundles compiled from the Motol study cohort. On first launch, the service layer detects the absence of user models and copies these bundles from the application's packaged resources into the user's local data directory, registering them with an `is_bundled` flag. Clinicians can therefore generate prognostic scores immediately after installation, without performing a training step, while retaining the ability to retrain on their own expanding cohort and promote the resulting local model to active status.

5.5.1 Build and Release

The project utilizes a CI pipeline via GitHub Actions to automate the creation of production-ready installers. While we had previously established the foundational pipeline for the Electron application, we extended it to accommodate new machine learning requirements and formalize the versioning process.

Automated ML Engine Compilation

A critical addition to the build process is the automated compilation of the Python analytical core. Currently, the pipeline exclusively targets a Windows release. On unsupported operating systems (such as macOS or Linux), the analytical sidecar

gracefully skips spawning, allowing the core database application to function without predictive capabilities. To ensure the correct ML engine bundling within the final Windows application, we implemented a dedicated build stage that executes on a `windows-latest` runner.

As shown in the pipeline configuration (see Listing 5.10), this stage installs the necessary scientific dependencies and utilizes `PyInstaller` to "freeze" the Python source into a standalone executable.

Listing 5.10: Automated Build of the Python ML Engine

```
C:\CSGDB>cd python-ml-engine
C:\CSGDB\python-ml-engine>pip install -r requirements.txt
C:\CSGDB\python-ml-engine>pip install pyinstaller
C:\CSGDB\python-ml-engine>./build.sh
C:\CSGDB\python-ml-engine>cd ..
C:\CSGDB>mkdir ml_engine
C:\CSGDB>mv python-ml-engine/dist/ml_engine.exe ml_engine/
```

This automation ensures that the most recent version of the prognostic logic is always encapsulated within the `ml_engine/` directory before the Electron packaging begins. By scripting this process, we eliminated the risk of manual bundling errors that could lead to version mismatches between the UI and the underlying analytical model. At runtime, the backend resolves the binary path dynamically via the Electron `app.isPackaged` property: in development, the engine is loaded from its local build output directory, while in production, it is read from the `extraResources` folder bundled by Electron Forge. This pathing strategy ensures that the packaged installer remains entirely self-contained.

Versioning and Release Management

To improve the reliability of clinical deployments, we implemented a strict versioning and release-gate system. Creation of a git tag (e.g., `v1.2.0`) triggers a pipeline that runs a validation check to ensure that a corresponding release notes file exists in the repository. This mechanism prevents the accidental publication of "blind" releases. Only when the release notes are verified does the pipeline proceed to create a formal GitHub Release. This tag versioning provides a clear audit trail for clinical users, allowing them to review changes in prognostic logic or UI updates before downloading the new installer.

Validation and System Evaluation

6

This chapter provides an evaluation of the prognostic module and its technical integration within the Czech Salivary Gland Database environment. We divided the validation process into two primary domains: the statistical assessment of the prognostic models and the verification of the software's operational integrity.

The first objective is to quantify the predictive accuracy and clinical reliability of the Random Survival Forest and Cox Proportional Hazards models. Given the inherent challenges of a low-event clinical dataset ($N = 122$), we utilize the .632 bootstrap estimator to mitigate optimistic bias and provide stable estimates of model discrimination and calibration. This section further includes a sensitivity analysis to justify the feature selection logic and ensure model parsimony.

The second objective addresses the technical robustness of the implementation. We detail a three-tier testing hierarchy—comprising unit, integration, and behavioral tests—designed to ensure data contract integrity across the Inter-Process Communication bridge. The chapter concludes with a synthesis of system performance metrics and formal clinical feedback from the Motol University Hospital, evaluating the system's readiness for real-world clinical decision support.

6.1 Model Validation

This section documents the statistical validation of the prognostic models integrated into the CSGDB application. We evaluate the performance of the RSF and Cox PH models using a retrospective dataset to verify their discriminative accuracy and clinical reliability.

Given the constraints of a localized oncological database ($N = 122$), our validation strategy focuses on mitigating optimistic bias and ensuring mathematical stability. We utilize the .632 bootstrap estimator to provide a robust assessment of internal validity. The following subsections detail quantitative performance across primary and secondary endpoints, model assumption verification, and the clinical alignment of the derived feature importance.

6.1.1 Predictive Performance Results

We evaluated the predictive performance using a retrospective cohort of $N = 122$ patients from the Czech Salivary Gland Database. The primary endpoint was RFS ($n = 25$ events), and the secondary endpoint was OS ($n = 33$ events). We used seven prognostic factors: age, number of positive nodes, T-stage, grade, lymphatic invasion, perineural invasion, and extranodal extension.

Model Discrimination and Accuracy

Table 6.1 summarizes the performance of the four models utilizing the .632 bootstrap estimator over 200 iterations. This methodology ensures that the reported metrics are corrected for the optimistic bias inherent in small-sample clinical datasets.

Table 6.1: Validation Results for RFS and OS Models (.632 Bootstrap).

Metric	RFS: RSF	RFS: Cox PH	OS: RSF	OS: Cox PH
C-index	0.814 ± 0.067	0.783 ± 0.087	0.761 ± 0.061	0.725 ± 0.075
IBS	0.115 ± 0.044	0.137 ± 0.063	0.154 ± 0.054	0.169 ± 0.061
Intercept	0.503 ± 0.963	−0.411 ± 1.182	—	—
Slope	1.485 ± 0.563	0.639 ± 0.487	—	—

The RSF achieved higher point estimates of discriminative performance for both endpoints. The recurrence RSF model achieved a C-index of 0.814, compared with 0.783 for the semi-parametric Cox PH baseline; however, the corresponding bootstrap intervals overlap, so the difference does not constitute a definitive demonstration of statistical superiority. The Integrated Brier Score (IBS) for both models remained substantially below the null-model threshold of 0.25, indicating predictive gain over a covariate-free baseline.

As illustrated in Figure 6.1, the Brier Score remains stable across the follow-up period. However, the confidence intervals widen at later time points due to a reduction in the number of patients at risk.

Calibration Analysis

Weak calibration was assessed at a one-year horizon ($t^* = 365$ days). The RFS models exhibited significant variance in calibration intercepts and slopes, a direct consequence of the low event count ($n = 25$). The RSF model showed an underconfident slope of 1.485, whereas the Cox PH model demonstrated a degree of overfitting with a slope of 0.639.

Regarding the OS models, calibration at the one-year horizon was omitted. Quasi-complete separation occurred in the logistic regression because the predicted probability of death at 365 days was extremely low across the cohort, reflecting the

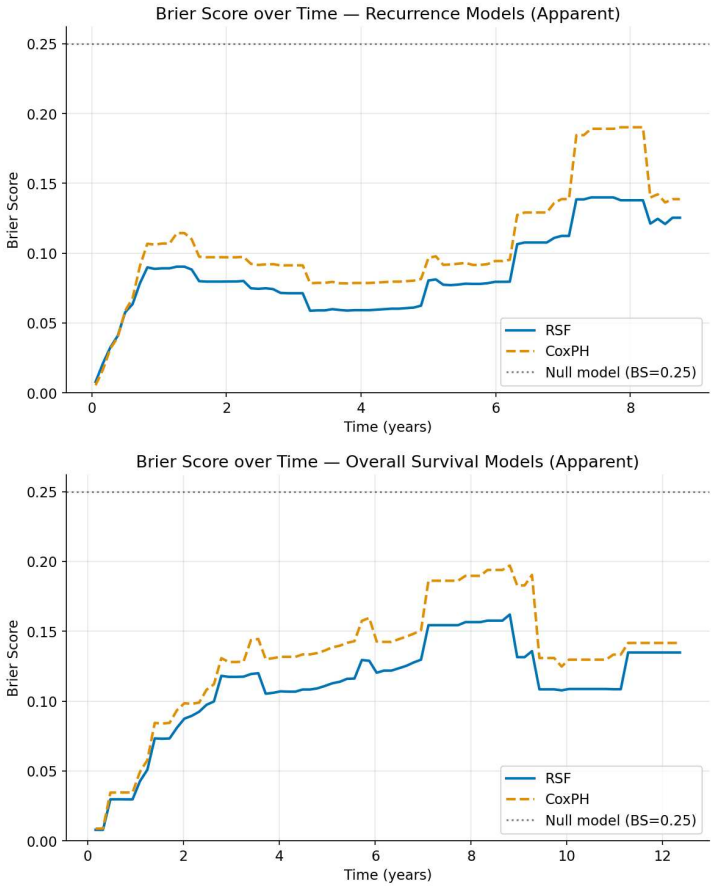


Figure 6.1: Brier curves for recurrence free and overall survival prediction.

favorable short-term prognosis of salivary gland malignancies. Consequently, calibration for OS was deemed numerically unstable for this specific temporal horizon.

6.1.2 Assumption Verification and Feature Importance

To ensure the structural integrity of our predictive framework, we verified the underlying mathematical assumptions of the Cox PH models and quantified the contribution of individual variables within the RSF ensembles.

Proportional Hazards Assumption

For the primary recurrence Cox PH model, we tested the PH assumption using Schoenfeld residuals against rank-transformed time. This step is critical to verify that the hazard ratios associated with our predictors remain constant over the follow-up period. As shown in Table 6.2, all p-values exceeded the $\alpha = 0.05$ thresh-

old, indicating that the null hypothesis—that hazards are proportional—cannot be rejected.

Table 6.2: Schoenfeld Residual Analysis for the Recurrence Cox PH Model.

Predictor	p **(rank)**	p **(KM)**	**Result**
Age at diagnosis	0.3513	0.4508	✓ Pass
Positive node count	0.3393	0.4797	✓ Pass
T-stage	0.7556	0.8476	✓ Pass
Grade	0.7288	0.8471	✓ Pass
Lymphatic invasion	0.5594	0.6201	✓ Pass
Perineural invasion	0.5960	0.4363	✓ Pass
Extranodal extension	0.4655	0.5167	✓ Pass

We concluded that the Cox PH model is mathematically valid for this dataset. We did not perform residual testing for the secondary survival Cox PH model, as it serves as a comparative baseline rather than the primary subject of clinical validation.

Permutation Feature Importance

We utilized permutation importance to assess the influence of each covariate on the RSF models. This was calculated as the mean decrease in the C-index when a feature's values are randomly shuffled (10 repeats on training data). This method allows us to identify which clinical markers most heavily drive the individual risk scores.

As illustrated in Figure 6.2, importance is distributed across all features, suggesting that the models leverage the entire clinical profile rather than a single dominant predictor. In the Recurrence RSF model, age at diagnosis (27.7%) and lymphatic invasion (23.4%) were the most influential factors.

In the OS RSF model, the relative importance of age increased to 39.7%. We attribute this shift to the fact that OS accounts for all-cause mortality, where age-related comorbidities play a more significant prognostic role compared to cancer-specific recurrence. Pathological markers such as perineural invasion and positive node count also maintained significant weight, aligning with established clinical consensus in salivary gland malignancy prognosis. Conversely, extranodal extension showed minimal importance ($< 2\%$), which we interpret as a result of data sparsity in the current CSGDB cohort rather than a lack of biological significance.

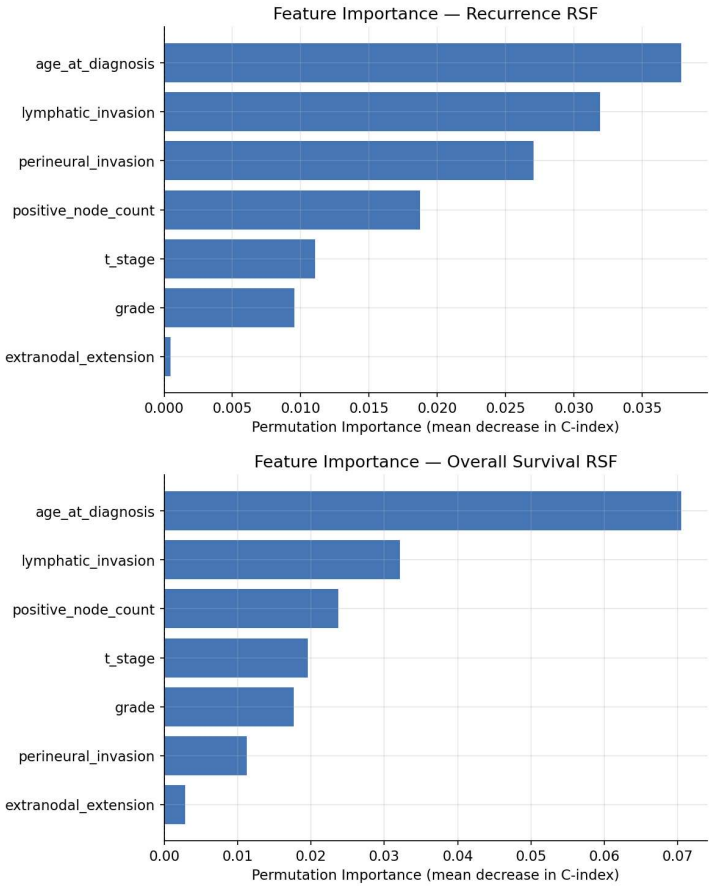


Figure 6.2: Feature importance recurrence free and overall survival prediction.

6.2 Sensitivity Analysis: Impact of Feature Set Complexity

To investigate the relationship between feature dimensionality and statistical stability, a three-way sensitivity analysis was conducted. We compared the primary 7-feature model against two parsimonious 4-feature configurations: a *Numeric* set (Age, Node count, T-stage, Grade) and a *Pathology* set (Node count, Lymphatic, Perineural, and Extranodal invasion).

This analysis serves two purposes: first, to identify the minimum viable prognostic signal required for accurate prediction; and second, to quantify how increasing the EPV ratio from 3.6 to 6.25 affects model calibration. We evaluated all models using the same 0.632 bootstrap methodology (200 iterations) as the primary analysis.

6.2.1 Recurrence-Free Survival: The Role of Invasion Markers

The validation of recurrence models (Table 6.3) reveals that the *Pathology* set is effectively equivalent to the 7-feature baseline in discriminative performance (C-index = 0.8132 vs. 0.8139). Conversely, the *Numeric* set showed a significant performance degradation (−0.033), suggesting that binary invasion markers carry the primary prognostic signal for tumor recurrence.

Table 6.3: Comparative Sensitivity Analysis: Feature Set Impact on Recurrence (RSF).

Configuration	C-index (0.632)	Slope (0.632)	EPV Ratio
7-Feature (Full)	0.8139 ± 0.067	1.4846 ± 0.563	3.57
4-Numeric set	0.7814 ± 0.076	1.7268 ± 0.286	6.25
4-Pathology set	**0.8132 ± 0.075**	**1.2448 ± 1.033**	6.25

The most significant finding is the improvement in the calibration slope for the *Pathology* set. By reducing the feature space, the RSF slope moved from 1.48 (underconfident) toward 1.24 (closer to the ideal 1.0). This empirical evidence confirms that the higher EPV ratio (6.25) reduces the over-parameterization bias observed in the primary model.

6.2.2 Overall Survival: Chronological Age as a Structural Anchor

The sensitivity analysis for OS underscores the indispensability of `age_at_diagnosis`. As summarized in Table 6.4, removing age (the *Pathology* set) resulted in a substantial decrease in the C-index (−0.047).

Furthermore, the OS calibration for the *Pathology* set exhibited total numerical collapse (Bootstrap $SD > 100$). We attribute this to the lack of a continuous "biological clock" feature, leaving the model unable to account for the high baseline mortality risk in the elderly population. This identifies chronological age as a structural prerequisite for stable OS modeling in salivary gland malignancies.

Table 6.4: Impact of Age Removal on Overall Survival Prediction (RSF).

Configuration	C-index (0.632)	IBS (0.632)	Stability
7-Feature (Full)	0.7607 ± 0.061	0.1535 ± 0.054	Stable
4-Numeric set	0.7595 ± 0.060	0.1643 ± 0.071	Stable
4-Pathology set	0.7135 ± 0.072	0.1601 ± 0.047	Unstable

6.2.3 Conclusion on Model Parsimony

The sensitivity analysis justifies the 7-feature model as the production standard. While the *Pathology* set offers superior calibration for recurrence, it lacks the stability required for OS estimation. The 7-feature model provides the most balanced performance across all endpoints. Moreover, it preserves clinically significant features like `extranodal_extension` which, despite their currently low statistical weight due to rarity, remain essential for clinical alignment as the database matures.

6.3 Methodological Limitations

The predictive performance reported in the preceding sections must be interpreted within the specific statistical and clinical boundaries of the current CSGDB cohort. Maintaining methodological rigor in oncology informatics requires a critical assessment of the factors that constrain the stability and generalizability of these models.

This section provides a transparent evaluation of the validation strategy's limitations. We specifically address the consequences of low event frequency on absolute risk precision and examine the trade-off between localized institutional optimization and global model transportability.

6.3.1 Data Sparsity and Estimator Stability

A fundamental constraint in this study is the suboptimal EPV ratio. For the primary recurrence endpoint ($n = 25$), the 7-feature model operates at an EPV of approximately 3.6, which is significantly below the threshold of $EPV \geq 10$ recommended by [48] for stable coefficient estimation.

The empirical consequences of this sparsity were quantified in the sensitivity analysis (Section 6.2). While reducing the feature set to four indicators improved the EPV to 6.25 and moved the calibration slope closer to unity (1.24 vs 1.48), the standard deviations remained high (± 1.03). This variance suggests that the *underconfidence* observed in the RSF models—where predicted risks are pulled toward the mean—is a direct result of the lack of event variance. Consequently, while the models demonstrate high discriminative ranking ability (C-index > 0.80), clinicians should interpret the absolute probability scores as indicative risk ranges rather than absolute certainties.

Furthermore, the "collapse" of the OS calibration in the 4-pathology configuration confirms a structural limitation: without chronological age as a continuous anchor, the model cannot converge on a stable mortality baseline. This identifies a data dependency in which clinical granularity (invasion markers) cannot compensate for missing fundamental demographic data in small-sample environments.

6.3.2 Validation Bias and Setting-Specific Overfitting

The findings remain inherently tied to the CSGDB cohort distribution. While the 0.632 bootstrap estimator was utilized to correct for internal optimistic bias, it cannot account for external "setting-specific" bias. The models are optimized for the clinical practice and pathological coding standards of Motol University Hospital. Institutional preferences—such as surgical margin protocols or specific radiotherapy escalation thresholds—may disproportionately influence model weights.

Without independent external validation against a geographically or chronologically distinct cohort, the universal transportability of these models remains unquantified. However, the primary architectural objective of this work was to enable localized clinical decision support. By prioritizing high performance within a specific institutional environment, the system remains responsive to local data distributions. While we caution against the extrapolation of these specific weights to international cohorts, the robust discrimination observed justifies the engine's utility as a specialized tool for the Czech Salivary Gland Database environment.

6.4 Software Verification and Performance

While the preceding sections validated the mathematical accuracy of the prognostic models, this section verifies the technical integrity of their implementation within the CSGDB application. We employ a multi-level testing hierarchy to ensure that the data pipeline, the IPC bridge, and the clinical logic remain robust across varying inputs and execution environments. Furthermore, we conduct a performance audit to quantify the system's responsiveness in a live clinical setting.

6.4.1 Three-Tier Testing Hierarchy

We implemented a three-tier testing suite utilizing the `pytest` framework to verify the software at different levels of granularity. This hierarchy ensures that both the atomic logic and the holistic system behavior adhere to clinical and technical requirements.

Tier 1: Unit Testing and Logic Isolation

The first tier focuses on isolating the internal logic of the feature extraction and validation modules. We utilized `test_feature_extractor.py` and `test_validators.py` to verify that clinical data from the SQLite database is correctly mapped to the model's input schema. Key verifications include the ordinal encoding of T-stage and Grade, the correct calculation of survival and recurrence

times from date strings, and the enforcement of the JSON data contract for all incoming requests.

Tier 2: Integration and IPC Integrity

The second tier, implemented in `test_ml_engine.py`, verifies the integration between the Electron host and the Python environment. These tests simulate the production lifecycle by asserting on the JSON responses across the IPC bridge. We verify that the `train`, `predict`, and `info` modes correctly handle data serialization and return appropriate error flags when encountering insufficient training data or missing model files, ensuring the host application can gracefully handle engine-level exceptions.

Tier 3: Behavioral Sanity and Clinical Plausibility

The final tier consists of behavioral tests in `sanity_check.py`. We used synthetic cohorts to verify that the model's outputs align with fundamental oncological principles, including the assertion that high-risk profiles consistently receive higher risk scores than low-risk profiles. Furthermore, we enforce **risk monotonicity**, ensuring that the predicted 5-year recurrence probability is at least as large as the 1-year probability, thereby preventing the generation of clinically impossible survival curves.

6.4.2 System Performance Audit

We conducted a performance audit to quantify the computational efficiency of the integrated machine learning engine within the CSGDB environment. This evaluation focused on the two primary execution paths: the periodic model retraining cycle and the real-time inference phase.

The training process, which involves executing 200 bootstrap iterations for the .632 estimator, requires approximately 60 seconds to complete. Given the database's retrospective nature and the relative infrequency of large-scale data updates, this latency is considered negligible within the administrative clinical workflow. Conversely, the inference phase is effectively instantaneous for the end-user.

6.5 Clinical Feedback

The clinical utility of the integrated predictive module was evaluated by a senior clinician at the Department of Otorhinolaryngology and Head and Neck Surgery, 1st Faculty of Medicine, University Hospital Motol. The assessment focused on the

operational integration of the RSF and Cox Proportional Hazards models within the diagnostic workflow of the CSGDB.

The formal institutional assessment is translated as follows:

> *"The newly implemented 'Machine Learning Risk Prediction' module within the Czech Salivary Gland Database represents a significant expansion of the system's functional capabilities toward advanced clinical decision support. The module utilizes a combination of RSF and Cox models to analyze two primary endpoints: overall survival and recurrence risk. Model training is performed on real-world, continuously updated patient data, enhancing the clinical relevance and local calibration of the outputs. Consequently, the system enables the estimation of individual risk profiles for newly enrolled patients, facilitating direct application in prognostic stratification, follow-up planning, and the selection of therapeutic strategies. A critical advantage of the architecture is the capacity for iterative retraining as new data or outcome parameters are recorded. This dynamic approach establishes the module as a viable tool for implementing personalized medicine in the domain of salivary gland malignancies."*

The original Czech documentation of this assessment is provided in Appendix D.

Discussion 7

The Random Survival Forest achieved a concordance index of 0.814 for recurrence-free survival and 0.761 for overall survival, representing a modest improvement over the semi-parametric Cox model. However, the overlapping confidence intervals derived from the .632 bootstrap estimator—0.814 ± 0.067 for the forest versus 0.783 ± 0.087 for the Cox model—preclude a definitive claim of statistical superiority. Both point estimates exceed those reported by comparable prognostic studies in salivary gland carcinoma: [50] reported a best-model C-index of 0.76 on a Danish parotid cohort of 726 patients, while the pre- and postoperative nomograms validated by [51] achieved C-indices of 0.74 and 0.71, respectively, on a cohort of 168 parotid cancer patients. The Peeperkorn cohort is the most directly comparable in scale to the present study ($N = 122$), while the Westergaard-Nielsen analysis provides a higher-power reference. The prognostic factors incorporated in the present feature set are consistent with those validated in [49], though that study did not report discriminative metrics. Two caveats temper the apparent advantage of the integrated models: first, the cited benchmarks were derived from parotid-only cohorts, whereas the Czech Salivary Gland Database encompasses all major salivary gland sites; and second, the modest absolute improvement is consistent with the optimism that small-cohort estimates retain even after the .632 bootstrap correction.

Analysis of model calibration identifies structural weaknesses in absolute risk estimation that warrant clinical caution. The Cox proportional hazards slope of 0.639 characterizes systematic overfitting, likely driven by the low event-per-variable ratio. In contrast, the Random Survival Forest slope of 1.485 indicates an under-confident model in which individual predictions are shrunk toward the cohort mean. Clinically, this implies that the system reliably ranks patient risk relative to the cohort but should not be interpreted as a calculator of absolute survival probabilities. The deployed user interface reflects this distinction: the `PatientRiskCard` foregrounds the relative risk score and the top three contributing factors, presenting the 1-, 3-, and 5-year probabilities as supporting context rather than as definitive point predictions.

At the feature level, the negligible permutation importance of extranodal ex-

tension suggests a lack of statistical signal due to data sparsity in this institutional cohort. This hypothesis should be revisited as the database matures. The sensitivity analysis in Section 6.2 further confirms that chronological age is a structural prerequisite for numerical stability: its removal leads to total calibration collapse in overall survival estimation. This bifurcation is clinically interpretable: biological aging governs cumulative all-cause mortality, whereas histopathological invasion drives tumor-specific recurrence, thereby justifying the dual-endpoint design of the integrated module.

Architecturally, the sidecar pattern achieves technical sovereignty and compliance with the General Data Protection Regulation via a local inter-process communication bridge, eliminating privacy risks associated with cloud-based inference services. The cost of this isolation is twofold: an increased installer footprint due to the bundled PyInstaller binary, and a non-trivial cold-start latency on first launch. However, the latter is amortized by the persistent sidecar process maintained throughout the clinical session, while transient processes are reserved for one-shot training jobs. The resulting decoupled infrastructure supports localized execution and asynchronous model retraining without compromising patient data integrity or clinical workflow efficiency at Motol University Hospital.

Conclusion 8

This thesis documents the fulfillment of all assigned objectives, transitioning the Czech Salivary Gland Database from a retrospective data-collection tool into a dynamic clinical decision-support system. While initial iterations—a semester project expanded during a bachelor's thesis—focused primarily on descriptive and inferential statistics, the current work integrates advanced machine learning to provide clinicians with individualized prognostic scores.

Integrating these prognostic capabilities required dismantling the legacy system's technical debt. We introduced a normalized Entity-Relationship Attribute model and broke the tight coupling between the user interface and persistence layer, building a stable infrastructure for reliable machine learning. Our design uses a sidecar architecture to offload survival calculations to a standalone Python core via a JSON IPC contract. This keeps the application scalable while ensuring sensitive patient records stay strictly local and secure on the physician's workstation.

Validation results indicate that the integrated models meet established performance standards for prognostic survival modeling in salivary gland carcinoma. The Random Survival Forest recorded a C-index of 0.814 for recurrence-free survival, against 0.783 for the Cox Proportional Hazards baseline; the overlapping bootstrap intervals temper this as a modest improvement rather than a definitive demonstration of superiority. Working with a small cohort of $N = 122$ with few observed events required careful handling of the low event-per-variable ratio. We relied on the .632 bootstrap estimator to correct for the optimistic bias inherent in small-sample evaluation, acknowledging that external validation on an independent cohort remains a prerequisite for transportability claims.

The software is now active at Motol University Hospital, where institutional assessment suggests it is a practical addition to risk stratification and treatment planning. Future updates should focus on research flexibility, such as allowing clinicians to select their own feature sets for modeling experiments. Another high-impact path is multimodal data integration, merging clinical parameters with features extracted from medical imaging. This project provides the sovereign, secure foundation needed to bring data science into the daily workflow of salivary gland oncology.

Original Database Full Table Columns

A

A.1 Malignant tumors of the submandibular gland

Table A.1: Malignant tumors of the submandibular gland.

Column Name	Type	Notes
id	int	pk, not null
form_type	int	not null
jmeno	text	
prijmeni	text	
id_pacient	text	
rodne_cislo	text	
vek_pri_diagnoze	int	
pohlavi	text	
kraj	text	
jine_nadorove_onemocneni_v_oa	text	
specifikace_mista_vyskytu_jineho_karcinomu	text	
jine_onemocneni_velkych_slinnych_zlaz_v_oa	text	
specifikace_onemocneni	text	
koureni	text	
pocet_cigaret_denne	int	
jak_dlouho_kouri	int	
pocet_balickoroku	real	
abusus_alkoholu	text	
rok_diagnozy	text	
diagnoza_mkn_10	text	
strana_nalezu	text	

Continues on next page

Table A.1 continued from previous page

Column Name	Type	Notes
funkce_n_vii_dle_h_b_predoperacne	text	
diagnosticke_metody	text	
fnab	text	
vysledek_fnab	text	
core_biopsie	text	
core_vysledek	text	
mukoepidermoidni_karcinom_core	text	
adenoidne_cysticky_karcinom_core	text	
polymorfni_adenokarcinom_core	text	
intraduktalni_karcinom_core	text	
salivarni_karcinom_nos_core	text	
karcinom_z_pleomorfniho_adenomu_core	text	
spatne_diferencovany_karcinom_core	text	
otevrena_biopsie	text	
otevrena_vysledek	text	
mukoepidermoidni_karcinom_otevrena	text	
adenoidne_cysticky_karcinom_otevrena	text	
polymorfni_adenokarcinom_otevrena	text	
intraduktalni_karcinom_otevrena	text	
salivarni_karcinom_nos_otevrena	text	
karcinom_z_pleomorfniho_adenomu_otevrena	text	
spatne_diferencovany_karcinom_otevrena	text	
datum_zahajeni_lecby	text	
typ_terapie	text	
rozsah_chirurgicke_lecby	text	
blokova_krcni_disekce	text	
strana_blokove_krcni_disekce	text	
typ_nd	text	
rozsah_nd	text	
funkce_n_vii_dle_h_b_pooperacne	text	
adjuvantni_terapie	text	
typ_nechirurgicke_terapie	text	
histopatologie_vysledek	text	
mukoepidermoidni_karcinom_histopatologie	text	
adenoidne_cysticky_karcinom_histopatologie	text	

Continues on next page

Table A.1 continued from previous page

Column Name	Type	Notes
polymorfni_adenokarcinom_histopatologie	text	
intraduktalni_karcinom_histopatologie	text	
salivarni_karcinom_nos_histopatologie	text	
karcinom_z_pleomorfniho_adenomu_histopatologie	text	
spatne_diferencovany_karcinom_histopatologie	text	
velikost_nadoru_histopatologie	int	
velikost_nadoru_neurcena_histopatologie	text	
okraj_resekce_histopatologie	text	
lymfovaskularni_invaze_histopatologie	text	
perineuralni_invaze_histopatologie	text	
pocet_lymfatickych_uzlin_s_metastazou_histopatologie	text	
extranodalni_sireni_histopatologie	text	
extranodalni_sireni_vysledek_histopatologie	text	
prokazane_vzdalene_metastazy_histopatologie	text	
misto_vyskytu_vzdalene_metastazy_histopatologie	text	
t_klasifikace_klinicka	text	
n_klasifikace_klinicka	text	
m_klasifikace_klinicka	text	
tnm_klasifikace_klinicka	text	
t_klasifikace_patologicka	text	
n_klasifikace_patologicka	text	
m_klasifikace_patologicka	text	
tnm_klasifikace_patologicka	text	
datum_prvni_kontroly_po_lecbe	text	
perzistence	text	
datum_prokazani_perzistence	text	
recidiva	text	
datum_prokazani_recidivy	text	
stav	text	
datum_umrti	text	
posledni_kontrola	text	
planovana_kontrola	text	
attachments	text	
poznamky	text	

A.2 Malignant tumors of the sublingual gland

Table A.2: Malignant tumors of the sublingual gland.

Column Name	Type	Notes
id	int	pk, not null
form_type	int	not null
jmeno	text	
prijmeni	text	
id_pacient	text	
rodne_cislo	text	
vek_pri_diagnoze	int	
pohlavi	text	
kraj	text	
jine_nadorove_onemocneni_v_oa	text	
specifikace_mista_vyskytu_jineho_karcinomu	text	
jine_onemocneni_velkych_slinnych_zlaz_v_oa	text	
specifikace_onemocneni	text	
koureni	text	
pocet_cigaret_denne	int	
jak_dlouho_kouri	int	
pocet_balickoroku	real	
abusus_alkoholu	text	
rok_diagnozy	text	
diagnoza_mkn_10	text	
strana_nalezu	text	
diagnosticke_metody	text	
fnab	text	
vysledek_fnab	text	
core_biopsie	text	
core_vysledek	text	
mukoepidermoidni_karcinom_core	text	
adenoidne_cysticky_karcinom_core	text	
polymorfni_adenokarcinom_core	text	
intraduktalni_karcinom_core	text	
salivarni_karcinom_nos_core	text	

Continues on next page

Table A.2 continued from previous page

Column Name	Type	Notes
karcinom_z_pleomorfniho_adenomu_core	text	
spatne_diferencovany_karcinom_core	text	
otevrena_biopsie	text	
otevrena_vysledek	text	
mukoepidermoidni_karcinom_otevrena	text	
adenoidne_cysticky_karcinom_otevrena	text	
polymorfni_adenokarcinom_otevrena	text	
intraduktalni_karcinom_otevrena	text	
salivarni_karcinom_nos_otevrena	text	
karcinom_z_pleomorfniho_adenomu_otevrena	text	
spatne_diferencovany_karcinom_otevrena	text	
datum_zahajeni_lecby	text	
typ_terapie	text	
rozsah_chirurgicke_lecby	text	
blokova_krcni_disekce	text	
strana_blokove_krcni_disekce	text	
typ_nd	text	
rozsah_nd	text	
adjuvantni_terapie	text	
typ_nechirurgicke_terapie	text	
histopatologie_vysledek	text	
mukoepidermoidni_karcinom_histopatologie	text	
adenoidne_cysticky_karcinom_histopatologie	text	
polymorfni_adenokarcinom_histopatologie	text	
intraduktalni_karcinom_histopatologie	text	
salivarni_karcinom_nos_histopatologie	text	
karcinom_z_pleomorfniho_adenomu_histopatologie	text	
spatne_diferencovany_karcinom_histopatologie	text	
velikost_nadoru_histopatologie	int	
velikost_nadoru_neurcena_histopatologie	text	
okraj_resekce_histopatologie	text	
lymfovaskularni_invaze_histopatologie	text	
perineuralni_invaze_histopatologie	text	
pocet_lymfatickych_uzlin_s_metastazou_histopatologie	text	
extranodalni_sireni_histopatologie	text	

Continues on next page

Table A.2 continued from previous page

Column Name	Type	Notes
extranodalni_sireni_vysledek_histopatologie	text	
prokazane_vzdalene_metastazy_histopatologie	text	
misto_vyskytu_vzdalene_metastazy_histopatologie	text	
t_klasifikace_klinicka	text	
n_klasifikace_klinicka	text	
m_klasifikace_klinicka	text	
tnm_klasifikace_klinicka	text	
t_klasifikace_patologicka	text	
n_klasifikace_patologicka	text	
m_klasifikace_patologicka	text	
tnm_klasifikace_patologicka	text	
datum_prvni_kontroly_po_lecbe	text	
perzistence	text	
datum_prokazani_perzistence	text	
recidiva	text	
datum_prokazani_recidivy	text	
stav	text	
datum_umrti	text	
posledni_kontrola	text	
planovana_kontrola	text	
attachments	text	
poznamky	text	

A.3 Malignant tumors of the parotid gland

Table A.3: Malignant tumors of the parotid gland.

Column Name	Type	Notes
id	int	pk, not null
form_type	int	not null
jmeno	text	
prijmeni	text	
id_pacient	text	
rodne_cislo	text	
vek_pri_diagnoze	int	

Continues on next page

Table A.3 continued from previous page

Column Name	Type	Notes
pohlavi	text	
kraj	text	
jine_nadorove_onemocneni_v_oa	text	
specifikace_mista_vyskytu_jineho_karcinomu	text	
jine_onemocneni_velkych_slinnych_zlaz_v_oa	text	
specifikace_onemocneni	text	
koureni	text	
pocet_cigaret_denne	int	
jak_dlouho_kouri	int	
pocet_balickoroku	real	
abusus_alkoholu	text	
rok_diagnozy	text	
diagnoza_mkn_10	text	
strana_nalezu	text	
funkce_n_vii_dle_h_b_predoperacne	text	
diagnosticke_metody	text	
fnab	text	
vysledek_fnab	text	
core_biopsie	text	
core_vysledek	text	
mukoepidermoidni_karcinom_core	text	
adenoidne_cysticky_karcinom_core	text	
polymorfni_adenokarcinom_core	text	
intraduktalni_karcinom_core	text	
salivarni_karcinom_nos_core	text	
karcinom_z_pleomorfniho_adenomu_core	text	
spatne_diferencovany_karcinom_core	text	
otevrena_biopsie	text	
otevrena_vysledek	text	
mukoepidermoidni_karcinom_otevrena	text	
adenoidne_cysticky_karcinom_otevrena	text	
polymorfni_adenokarcinom_otevrena	text	
intraduktalni_karcinom_otevrena	text	
salivarni_karcinom_nos_otevrena	text	
karcinom_z_pleomorfniho_adenomu_otevrena	text	

Continues on next page

Table A.3 continued from previous page

Column Name	Type	Notes
spatne_diferencovany_karcinom_otevrena	text	
datum_zahajeni_lecby	text	
typ_terapie	text	
rozsah_chirurgicke_lecby	text	
blokova_krcni_disekce	text	
strana_blokove_krcni_disekce	text	
typ_nd	text	
rozsah_nd	text	
funkce_n_vii_dle_h_b_pooperacne	text	
pooperacni_komplikace	text	
jine_pooperacni_komplikace	text	
adjuvantni_terapie	text	
typ_nechirurgicke_terapie	text	
histopatologie_vysledek	text	
mukoepidermoidni_karcinom_histopatologie	text	
adenoidne_cysticky_karcinom_histopatologie	text	
polymorfni_adenokarcinom_histopatologie	text	
intraduktalni_karcinom_histopatologie	text	
salivarni_karcinom_nos_histopatologie	text	
karcinom_z_pleomorfniho_adenomu_histopatologie	text	
spatne_diferencovany_karcinom_histopatologie	text	
velikost_nadoru_histopatologie	int	
velikost_nadoru_neurcena_histopatologie	text	
okraj_resekce_histopatologie	text	
lymfovaskularni_invaze_histopatologie	text	
perineuralni_invaze_histopatologie	text	
pocet_lymfatickych_uzlin_s_metastazou_histopatologie	text	
extranodalni_sireni_histopatologie	text	
extranodalni_sireni_vysledek_histopatologie	text	
prokazane_vzdalene_metastazy_histopatologie	text	
misto_vyskytu_vzdalene_metastazy_histopatologie	text	
t_klasifikace_klinicka	text	
n_klasifikace_klinicka	text	
m_klasifikace_klinicka	text	
tnm_klasifikace_klinicka	text	

Continues on next page

Table A.3 continued from previous page

Column Name	Type	Notes
t_klasifikace_patologicka	text	
n_klasifikace_patologicka	text	
m_klasifikace_patologicka	text	
tnm_klasifikace_patologicka	text	
datum_prvni_kontroly_po_lecbe	text	
perzistence	text	
datum_prokazani_perzistence	text	
recidiva	text	
datum_prokazani_recidivy	text	
stav	text	
datum_umrti	text	
posledni_kontrola	text	
planovana_kontrola	text	
attachments	text	
poznamky	text	

A.4 Benign tumors of the submandibular gland

Table A.4: Benign tumors of the submandibular gland.

Column Name	Type	Notes
id	int	pk, not null
form_type	int	not null
jmeno	text	
prijmeni	text	
id_pacient	text	
rodne_cislo	text	
vek_pri_diagnoze	int	
pohlavi	text	
kraj	text	
jine_nadorove_onemocneni_v_oa	text	
specifikace_mista_vyskytu_jineho_karcinomu	text	
jine_onemocneni_velkych_slinnych_zlaz_v_oa	text	
specifikace_onemocneni	text	

Continues on next page

Table A.4 continued from previous page

Column Name	Type	Notes
koureni	text	
pocet_cigaret_denne	int	
jak_dlouho_kouri	int	
pocet_balickoroku	real	
abusus_alkoholu	text	
rok_diagnozy	text	
strana_nalezu	text	
funkce_n_vii_dle_h_b_predoperacne	text	
diagnosticke_metody	text	
fnab	text	
vysledek_fnab	text	
core_biopsie	text	
core_vysledek	text	
core_vysledek_jine	text	
otevrena_biopsie	text	
otevrena_vysledek	text	
otevrena_vysledek_jine	text	
datum_zahajeni_lecby	text	
typ_terapie	text	
rozsah_chirurgicke_lecby	text	
funkce_n_vii_dle_h_b_pooperacne	text	
pooperacni_komplikace	text	
jine_pooperacni_komplikace	text	
histopatologie_vysledek	text	
histopatologie_vysledek_jine	text	
velikost_nadoru_histopatologie	int	
velikost_nadoru_neurcena_histopatologie	text	
okraj_resekce_histopatologie	text	
datum_prvni_kontroly_po_lecbe	text	
doporuceno_dalsi_sledovani	text	
perzistence	text	
datum_prokazani_perzistence	text	
recidiva	text	
datum_prokazani_recidivy	text	
stav	text	

Continues on next page

Table A.4 continued from previous page

Column Name	Type	Notes
datum_umrti	text	
posledni_kontrola	text	
planovana_kontrola	text	
attachments	text	
poznamky	text	

A.5 Benign tumors of the parotid gland

Table A.5: Benign tumors of the parotid gland.

Column Name	Type	Notes
id	int	pk, not null
form_type	int	not null
jmeno	text	
prijmeni	text	
id_pacient	text	
rodne_cislo	text	
vek_pri_diagnoze	int	
pohlavi	text	
kraj	text	
jine_nadorove_onemocneni_v_oa	text	
specifikace_mista_vyskytu_jineho_karcinomu	text	
jine_onemocneni_velkych_slinnych_zlaz_v_oa	text	
specifikace_onemocneni	text	
koureni	text	
pocet_cigaret_denne	int	
jak_dlouho_kouri	int	
pocet_balickoroku	real	
abusus_alkoholu	text	
rok_diagnozy	text	
strana_nalezu	text	
funkce_n_vii_dle_h_b_predoperacne	text	
diagnosticke_metody	text	
fnab	text	
vysledek_fnab	text	

Continues on next page

Table A.5 continued from previous page

Column Name	Type	Notes
core_biopsie	text	
core_vysledek	text	
core_vysledek_jine	text	
otevrena_biopsie	text	
otevrena_vysledek	text	
otevrena_vysledek_jine	text	
datum_zahajeni_lecby	text	
typ_terapie	text	
rozsah_chirurgicke_lecby	text	
funkce_n_vii_dle_h_b_pooperacne	text	
pooperacni_komplikace	text	
jine_pooperacni_komplikace	text	
histopatologie_vysledek	text	
histopatologie_vysledek_jine	text	
velikost_nadoru_histopatologie	int	
velikost_nadoru_neurcena_histopatologie	text	
okraj_resekce_histopatologie	text	
datum_prvni_kontroly_po_lecbe	text	
doporuceno_dalsi_sledovani	text	
perzistence	text	
datum_prokazani_perzistence	text	
recidiva	text	
datum_prokazani_recidivy	text	
stav	text	
datum_umrti	text	
posledni_kontrola	text	
planovana_kontrola	text	
attachments	text	
poznamky	text	

CSGDB Data Dictionary

B

This appendix provides a comprehensive listing of all data fields available in the Czech Salivary Gland Database for malignant gland forms (Parotid, Submandibular, and Sublingual).

B.1 Personal Data (Anamnestic)

Table B.1: Personal and anamnestic data fields.

Attribute Name	Data Type	Options / Notes
First name	Text	Required, validated
Surname	Text	Required, validated
Patient ID	Text	Internal clinical identifier
Birth number	Text	Required, Czech PID format (without slash)
Age at diagnosis	Number	Calculated from birth number and diagnosis year
Gender	Single-select	Female, Male
Residency region	Single-select	14 Regions of the Czech Republic
Other cancer in OA	Yes / No	If Yes: specify location (text)
Other salivary gland disease in OA	Yes / No	If Yes: specify disease (text)
Smoking	Yes / No / Unk.	If Yes: cigarettes/day, duration (years), pack-years (calc.)
Alcohol abuse	Single-select	Yes, No, Unknown

B.2 Diagnosis and Diagnostic Methods

Table B.2: Diagnostic data fields.

Attribute Name	Data Type	Options / Notes
Year of diagnosis	Date	
Diagnosis ICD-10	Single-select	C07 (parotid), C08.0 (submand.), C08.1 (subling.)
Side of finding	Single-select	Right, Left, Bilateral
N VII function (pre-op)	Single-select	House-Brackmann scale I–VI (Parotid and Submand. only)
Diagnostic methods used	Multi-select	UZ, CT, MRI, PET-CT, PET-MR, No imaging
FNAB performed	Yes / No	
FNAB result	Single-select	MSRSGC classification (I–VI)
Core biopsy performed	Yes / No	
Core biopsy result	Single-select	See Tumor Type List (Table B.3)
Open biopsy performed	Yes / No	
Open biopsy result	Single-select	See Tumor Type List (Table B.3)

Table B.3: Standardized Tumor Type List.

Tumor Type	Subtype Options / Details
Acinic cell carcinoma	—
Secretory carcinoma	—
Mucoepidermoid carcinoma	Low, Intermediate, High grade, NOS
Adenoid cystic carcinoma	Tubular/cribriform, >30% solid, NOS
Polymorphous adenocarcinoma	Classic, Cribriform, NOS
Epithelial-myoepithelial ca.	—
Hyalinizing clear cell ca.	—
Basal cell adenocarcinoma	—
Sebaceous adenocarcinoma	—

Continues on next page

Table B.3 continued from previous page

Tumor Type	Subtype Options / Details
Intraductal carcinoma	Intercalated duct-like, Apocrine, Oncocytic, Mixed, NOS
Salivary carcinoma NOS	Oncocytic, Intestinal-type, NOS
Salivary duct carcinoma	—
Myoepithelial carcinoma	—
Ca. ex pleomorphic adenoma	Intracapsular, Minimally invasive, Invasive, NOS
Carcinosarcoma	—
Poorly differentiated ca.	Undifferentiated, Large cell neuro., Small cell neuro., NOS
Lymphoepithelial carcinoma	—
Squamous cell carcinoma	—
Microsecretory adeno-ca.	—
Sclerosing microcystic adeno-ca.	—
Mucinous adenocarcinoma	—
Sialoblastoma	—
MALT lymphoma	—
Other	Specific for Submandibular and Sublingual

B.3 Therapy

Table B.4: Therapeutic data fields.

Attribute Name	Data Type	Options / Notes
Treatment start date	Date	
Therapy type	Single-select	Surgical, Non-surgical, Not indicated
Surgical extent (Parotid)	Single-select	Parotidectomy types (I-VII), ECD
Surgical extent (Other)	Single-select	Peroral extirpation, External extirpation, Other
ND	Yes / No	
Side of ND	Single-select	Same side, Two-sided

Continues on next page

Table B.4 continued from previous page

Attribute Name	Data Type	Options / Notes
ND type	Single-select	Elective (cN0), Therapeutic (cN+)
ND scope	Multi-select	Levels Ia, Ib, IIa, IIb, III, IV, V, Va, Vb, VI
N VII function (post-op)	Single-select	House-Brackmann I–VI (Parotid and Submand. only)
Post-op complications	Single-select	None, Freyś syndrome, Salivary fistula, Other
Adjuvant therapy	Single-select	RT, ChRT, CH, Proton therapy, Not indicated
Oncological treatment	Single-select	For non-surgical: RT, ChRT, CH, Proton therapy

B.4 Histopathology and TNM Classification

Table B.5: Histopathological and staging fields.

Attribute Name	Data Type	Options / Notes
Histological type	Single-select	See Tumor Type List (Table B.3)
Tumor size	Number	Largest dimension in mm
Resection margin	Single-select	R0, R1
Lymphovascular invasion	Single-select	No, Yes
Perineural invasion	Single-select	No, Yes
Positive lymph nodes	Number	Absolute count of nodes with metastasis
ENE	Yes / No	If Yes: ENEma (>2 mm), ENEmi (≤2 mm)
Distant metastases	Yes / No	If Yes: specify site (text)
TNM edition	Toggle	Dynamically loaded from DB
Clinical T, N, M, Stage	Single-select	Based on selected TNM edition

Continues on next page

Table B.5 continued from previous page

Attribute Name	Data Type	Options / Notes
Pathological T, N, M, Stage	Single-select	Based on selected TNM edition

B.5 Follow-up and Outcomes

Table B.6: Follow-up data fields.

Attribute Name	Data Type	Options / Notes
First follow-up date	Date	Date of first checkup after treatment
Persistence	Yes / No / Unk.	If Yes: date of proven persistence
Recurrence	Yes / No / Unk.	If Yes: date of proven recurrence
Status	Single-select	Alive, Deceased, Unknown
Date of death	Date	If status is Deceased
Last checkup date	Date	
Scheduled checkup date	Date	
Attachments	File	Images, documents (as file uploads)
Notes	Text	Free-text multi-line field

IOS 1/2011-5 C

Fakultní nemocnice v Motole

V Úvalu 84, 150 06 Praha 5

FN MOTOL

Organizační směrnice č. IOS_1/2011-5

Klinický výzkum

Určení: všem zaměstnancům zdravotnických pracovišť Fakultní nemocnice v Motole.

	Zpracoval:	**Garant:**	**Schválil:**
Organizační útvar	Úsek náměstka pro vědu a výzkum	Úsek léčebně preventivní péče	Úsek ředitele
Funkce	Náměstek pro vědu a výzkum	Náměstek pro léčebně preventivní péči	Ředitel FN Motol
Jméno	Prof. MUDr. Anna Šedivá, DSc.	MUDr. Martin Holcát, MBA	JUDr. Ing. Miloslav Ludvík, MBA

Účinnost směrnice od: 1. 1. 2011
Účinnost této verze od: 1. 10. 2020
Doba platnosti: bez omezení, revize dle čl. 6.9 směrnice č. IOS_23/2006-5, ve znění pozdějších revizí
Verze: 5
Počet stran směrnice: 15
Počet příloh: 2
Umístění podepsaného výtisku: Ministerstvo zdravotnictví České republiky
Administrátor pro řízenou dokumentaci
Rozdělovník řízených výtisků: Právní odbor
Odbor vnitřního auditu a řízení kvality

List provedených revizí a změn

Změna č.	Umístění změny	Popis provedené změny	Datum účinnosti	Odpovědná osoba
1	průběžně	Kompletní přepracování	1.5.2013	Prof. MUDr. A. Šedivá
2	průběžně	Revize, změna Výzkumného záměru za Institucionální podporu výzkumné organizace	1.10.2014	Prof. MUDr. Anna Šedivá, DSc
3	průběžně	Celková revize, uvedení Agentury pro zdravotnický výzkum, ochrana duševního vlastnictví	1.3.2017	Prof. MUDr. Anna Šedivá, DSc
4	průběžně	Revize podle platné legislativy, včlenění GDPR pravidel	1. 10. 2020	Prof. MUDr. Anna Šedivá, DSc
5	Přílohy	Nové přílohy č. 1 a č. 2	1. 10. 2020	Prof. MUDr. Anna Šedivá, DSc

Obsah

Přílohy:

Příloha č. 1 Diagram – plánování projektu

Příloha č. 2 Diagram – správa dat

Při plánování projektu a správě dat v oblasti vědy a výzkumu se postupuje dle rozhodovací linie v přílohách č. 1 a 2. Ve specifických situacích, které svou výzkumnou povahou vybočují ze standardního postupu, se situace řeší individuálně na základě konzultace s pověřencem pro ochranu osobních údajů.

Klíčová slova:

klinický výzkum, informovaný souhlas pacienta, využití výsledku výzkumu, účelová a institucionální podpora výzkumu, výzkumný záměr, dotační programy EU a Norských fondů.

1 Účel

Účelem směrnice je stanovit podmínky ke sjednocení postupu při organizování klinického výzkumu ve FN Motol.

2 Rozsah platnosti

Všem zaměstnancům zdravotnických pracovišť FN Motol.

3 Definice a použité zkratky

3.1 Definice

Klinický výzkum je výzkumná činnost, která zahrnuje účast pacienta či dobrovolníků, ve výjimečných a jasně definovaných případech.

Informovaný souhlas – je projev vůle k účasti v klinickém výzkum, který má písemnou povahu, a ve kterém pacient či dobrovolník dobrovolně potvrzuje svou ochotu podílet se na konkrétním klinickém výzkumu poté, co byl informován o všech aspektech klinického výzkumu, které jsou důležité pro jeho rozhodnutí zúčastnit se klinického výzkumu.

Účastník výzkumu - pro účely této směrnice se rozumí pacient osobně nebo jeho zákonný zástupce.

3.2 Zkratky

AZV ČR	Agentura pro zdravotnický výzkum České republiky
ČSN ISO	Česká státní norma systému řízení jakosti
EU	Evropská Unie
FM EHP/Norsko	Finanční mechanismus evrop. hospodářského prostoru a Norska
FN Motol	Fakultní nemocnice v Motole
GAČR	Grantová agentura České republiky
GDPR	Nařízení Evropského parlamentu a Rady (EU) 2016/679 ze dne 27. dubna 2016 o ochraně fyzických osob v souvislosti se zpracováním osobních údajů a o volném pohybu těchto údajů a o zrušení směrnice 95/46/ES (obecné nařízení o ochraně osobních údajů)
IP FN Motol	Institucionální podpora Fakultní nemocnice v Motole
MZ ČR	Ministerstvo zdravotnictví České republiky
NKM	Národní kontaktní místo
NVV	Náměstek pro vědu a výzkum
RVVI	Rada pro výzkum, vývoj a inovace
SÚKL	Státní ústav pro kontrolu léčiv
TV	Televizní přijímač
USA	Spojené státy americké

4 Klinický výzkum ve FN Motol

4.1. FN Motol se podílí na rozvíjení vědy a výzkumu, na ověřování nových diagnostických a léčebných metod, provádění klinického hodnocení léčiv a nové zdravotnické techniky a zavádění výsledků vědy do praxe.

4.2. FN Motol ve vzdělávání, vědě a výzkumu a zdravotnictví úzce spolupracuje s univerzitními a akademickými pracovišti, resortními ústavy a dalšími odbornými organizacemi.

4.3. Klinický výzkum se uskutečňuje na základě níže uvedených zásad, které jsou pro všechny jeho oblasti společné. Pokud jsou jednotlivé oblasti či kategorie klinického výzkumu upraveny zvláštními interními opatřeními, která některá pravidla upravují podrobněji či odchylně, má tato zvláštní úprava přednost.

4.4. Veškerý zisk z vědecké a výzkumné činnosti bude zpětně investován do provádění výzkumu a vývoje, včetně šíření jejich výsledků.

4.5. Každý lékař FN Motol, který realizuje klinický výzkum, je povinen poskytnout účastníku výzkumu informace a poučení o možnosti účasti na klinickém výzkumu, pokud to vzhledem k jeho léčbě je vhodné, tj. zajistit informovaný souhlas účastníka výzkumu.

Tento informovaný souhlas zahrnuje zejména:

- upozornění, že se jedná o výzkumnou činnost a cíle klinického výzkumu,
- postupy a výkony v průběhu klinického výzkumu se zdůrazněním těch prvků klinického výzkumu, které mají povahu výzkumu,
- předvídatelná rizika či nepříjemnosti pro účastníky výzkumu,
- očekávané přínosy pro účastníky výzkumu,
- alternativní léčebné postupy, které mohou být pro léčbu účastníka výzkumu použity, jejich výhody a rizika,
- podmínky odškodnění v případě újmy na zdraví vzniklé v souvislosti s účastí v klinickém výzkumu,
- požadavky na spolupráci a léčebný režim účastníka výzkumu,
- informaci o tom, že účast subjektu v klinickém výzkumu je dobrovolná a že tento subjekt může svoji účast odmítnout nebo může odstoupit od účasti v klinickém výzkumu kdykoliv bez udání důvodů, bez postihu či ztrát výhod, na něž má jinak nárok,
- souhlas s tím, že oprávněné subjekty v klinickém výzkumu budou mít umožněn přístup k potřebným informacím za účelem ověření průběhu klinického výzkumu, aniž dojde k porušení důvěrnosti informací o účastnících klinického výzkumu, v míře povolené

právními předpisy, a že podepsáním písemného informovaného souhlasu účastník klinického výzkumu či jeho pověřený zákonný zástupce souhlasí s touto skutečností,

- souhlas s tím, že záznamy, podle nichž lze identifikovat účastníka výzkumu, budou uchovávány jako důvěrné a nebudou, v míře zaručené právními předpisy, veřejně zpřístupněny; budou-li výsledky klinického výzkumu publikovány, totožnost účastníka výzkumu nebude zveřejněna,
- souhlas s tím, že účastník výzkumu anebo jeho zákonný zástupce budou včas informováni, pokud by se vyskytla informace, která by mohla mít význam pro rozhodnutí účastníka klinického výzkumu pokračovat v jeho účasti,
- předvídatelné okolnosti a důvody, pro které může být účast subjektu v klinickém výzkumu ukončena,
- informace o osobách, od kterých bude možné získat další informace týkající se klinického výzkumu a práv subjektů klinického výzkumu,
- předpokládaná doba trvání účasti v klinickém výzkumu,
- informaci o tom, kdo je správcem osobních údajů, včetně sídla, IČO, případně dalších identifikačních údajů,
- informaci o tom, zda budou osobní údaje předávány jinému příjemci (tj. mimo Fakultní nemocnici v Motole), zda budou předávány mimo ČR a pokud ano, zda v rámci EU či mimo EU, popř. uvést kategorie příjemců osobních údajů,
- informaci o tom, jak dlouho budou osobní údaje archivovány (uloženy) a z jakého důvodu, případně kritéria pro určení této doby,
- informaci o tom, zda osobní údaje, které byly získány pro příslušný klinický výzkum, budou využity pro jiné (další) účely. Tyto účely musí být slučitelné s původním záměrem,
- informaci, že další informace o zpracování osobních údajů jsou dostupné v Prohlášení o zpracování osobních údajů (GDPR) dostupné na webových stránkách Fakultní nemocnice v Motole (www.fnmotol.cz).

Na podávání výše uvedených informací účastníkům hodnocení se přiměřeně použijí pravidla stanovená směrnicí č. IOS_10/2005-7 Informovaný souhlas pacienta, ve znění pozdějších revizí.

4.6. Každý lékař a zdravotnický pracovník FN Motol, který realizuje klinický výzkum, musí v jeho rámci primárně zabezpečit ochranu práva, bezpečnost a zdraví pacienta či dobrovolníka, neboť to jsou nejdůležitější hlediska a mají převažovat nad zájmy vědy a společnosti.

4.7. Každý lékař a zdravotnický pracovník FN Motol, který realizuje klinický výzkum, je povinen dodržovat platné právní předpisy s ohledem na dodržování mlčenlivosti a nakládání

s osobními údaji a osobními údaji zvláštní kategorie pacientů účastnících se klinického výzkumu, upravené zejména zákonem č. 110/2019 Sb., o zpracování osobních údajů, ve znění pozdějších předpisů, zákonem č. 372/2011 Sb., o zdravotních službách a podmínkách jejich poskytování, ve znění pozdějších předpisů a Nařízení Evropského parlamentu a Rady (EU) 2016/679 ze dne 27. dubna 2016 o ochraně fyzických osob v souvislosti se zpracováním osobních údajů a o volném pohybu těchto údajů a o zrušení směrnice 95/46/ES (obecné nařízení o ochraně osobních údajů).

5 Klinické hodnocení léčiv, klinické hodnocení zdravotnických prostředků a klinické zkoušky zdravotnických prostředků

5.1. Podrobný postup při provádění klinického hodnocení léčiv ve FN Motol je upraven organizační směrnicí č. IOS_3/2010-3 Klinické hodnocení, ve znění pozdějších revizí.

5.2. Etická komise FN Motol posuzuje projekty biomedicínckého výzkumu zahrnující lidské účastníky, schvaluje žádosti v oblasti klinického hodnocení léčiv, rovněž schvaluje žádosti v oblasti klinického hodnocení a klinických zkoušek zdravotnických prostředků, vyjadřuje se k ostatním výzkumným projektům, popř. k etickým otázkám souvisejícím s poskytováním zdravotních služeb pacientům Fakultní nemocnice v Motole.

6 Účelová a institucionální podpora výzkumu

6.1. Výzkum, který je předmětem grantových projektů, se řídí pravidly, která jsou stanovena poskytovatelem (MZČR, grantové agentury domácí i zahraniční, výjimečně další subjekty). Institucionální podpora je uvnitř FN Motol členěna do systému Interních grantů. Interní granty se řídí vlastními interními opatřeními, která jsou uvedena na intranetu FN Motol, v sekci směrnice-Věda, výzkum a výuka, organizační směrnice č. IOS_7/2002-4 Interní granty FN Motol.

6.2. Za administrativní stránku této části výzkumných aktivit odpovídá ekonomický odbor ve spolupráci se zaměstnaneckým odborem FN Motol.

6.3. Za vědeckou stránku výzkumné činnosti odpovídá úsek náměstka pro vědu a výzkum, vědečtí koordinátoři vybraných domácích či zahraničních grantů na základě sjednaných smluv.

6.4. Každý grantový či jiný výzkumný projekt podléhá při svém podání schválení Etickou komisí FN Motol, Vědeckou radou FN Motol a zároveň musí být splněny podmínky uvedené ve směrnici IOS_4/2018-1 Ochrana osobních údajů, ve znění pozdějších revizí. Zejména musí být zpracována potřebná dokumentace a předložena k vyjádření pověřenci

pro ochranu osobních údajů FN Motol. Proces posouzení se spouští vyplněním přílohy číslo 1 výše uvedené směrnice.

6.5. Informované souhlasy pacientů nebo zákonných zástupců nezletilých pacientů, účastnících se výzkumu, jsou nedílnou součástí grantové přihlášky, jsou-li potřeba.

6.6. Grantové přihlášky, které řeší projekty, jež jsou svou povahou klinickými studiemi, jsou dále postoupeny ke stanovisku Státnímu ústavu pro kontrolu léčiv (dále jen „SUKL“), prostřednictvím grantového oddělení FN Motol, což umožňuje dodržovat zákonem stanovená pravidla při diagnostickém a léčebném využití klasických i nových metod včetně buněčné terapie či využití tkáňových náhrad. V případě, že řešení grantu podléhá schválení SUKLu a grantu je přidělená účelová podpora, zajistí hlavní řešitel před uzavřením smlouvy požadavky stanovené výše uvedenou institucí. Výsledek stanoviska je předán grantové agentuře.

Vědecká rada dále kontroluje průběžné zprávy grantových projektů, které jsou pravidelně podávány v ročních vyhlášených termínech grantových agentur.

Institucionální forma výzkumu je realizována prostřednictvím Institucionální podpory na dlouhodobý koncepční rozvoj výzkumné organizace na základě zhodnocení jí dosažených výsledků, přidělené FN Motol rozhodnutím Ministerstva zdravotnictví České republiky. Využití Institucionální podpory je spravováno Vědeckou radou FN Motol a náměstkyní pro vědu a výzkum a řídí se směrnicí č. IOS_7/2002-4 Interní granty FN Motol.

6.7. Administrativně je Institucionální podpora spravována ekonomickým odborem ve spolupráci se zaměstnaneckým odborem FN Motol podle pravidelně aktualizovaných podmínek poskytovatele.

7 Moderní terapie ve FN Motol

7.1. Systém Moderních terapií je představován podporou progresivních nových postupů s potenciálem překlenutí výzkumu do klinického použití. Tento systém je podporován vlastními prostředky FN Motol.

7.2. Do skupiny projektů pojatých jako Moderní terapie spadá výzkum inovativních postupů v oblasti buněčných terapií, tkáňových náhrad, nových diagnostických a léčebných postupů a zavádění dalších progresivních metod, jejichž výstupem je získání klinické studie, užitného vzoru či patentu s následnou spoluprací s technologickými firmami či jiná forma praktického uplatnění.

7.3. Výzkumné projekty Moderních terapií jsou vybírány a sledovány Vědeckou radou FN Motol. Zajištění jejich ekonomické a personální stránky je prováděno obdobně jako u ostatních grantů ekonomickým a zaměstnaneckým odborem FN Motol.

7.4. Administrativní zajištění externích dokumentů a povolení k takto definovaným výzkumným postupům a k jejich následnému praktickému uplatnění je v kompetenci Úseku pro vědu a výzkum a řeší se individuálně podle povahy projektu a platných odpovídajících regulačních pravidel.

8 Finanční podpora evropských a ostatních fondů (granty EU a Norské fondy)

8.1. Výzkum, který je realizovaný z finanční podpory grantů EU, evropských a ostatních fondů (např. FM EHP/Norsko), se řídí pravidly jednotlivých fondů a čerpání takto poskytnutých prostředků probíhá v souladu s příručkou a s pokyny řídícího orgánu operačního programu /Národního kontaktního místa (NKM) v případě Norských fondů/ k danému projektu. Totéž platí i pro granty s bilaterální mezinárodní účastí v rámci GAČR, či pro granty získané od jiných zahraničních poskytovatelů, eventuálně i pro ostatní zahraniční grantové agentury.

8.2. Projekty EU jsou řízeny Vědeckou radou FN Motol a administrativně spravovány Úsekem pro vědu a výzkum, Oddělením strukturálních fondů EU a jiných dotačních programů. Projekty jsou interně řízeny v souladu s prováděcími předpisy dozorových orgánů a institucí, zejména s metodickou Příručkou každého projektu, Pokyny k vypracování a podání projektové žádosti, s Podmínkami rozhodnutí o přidělení dotace, Závaznými postupy a termíny pro zahájení a realizaci projektu, náležitostmi Žádosti o platbu a Pokyny k vyplnění monitorovacích zpráv a periodických hlášení, společně s příslušnými smlouvami a s operativními pokyny řídícího orgánu operačního programu (nebo NKM). U těchto projektů se podle výše uvedené dokumentace zhotovují pravidelná etapová hlášení a pravidelné etapové monitorovací zprávy. V době ukončení projektu je zpracována závěrečná monitorovací zpráva o realizaci projektu a po dobu udržitelnosti projektu se zasílají další monitorovací zprávy. Financování projektu probíhá podle platných pravidel daného schématu EU projektů konkrétně pro každý přiznaný projekt.

9 Výzkum nových metod a léčebných postupů

9.1. Veškeré nové léčebné a diagnostické postupy, v rámci grantových projektů domácích, zahraničních, včetně Institucionální podpory Fakultní nemocnice v Motole, pro jejich klinické využití vyžadují dodržení směrnice č. IOS_3/2004-4 Zavedení nového léčebného postupu, ve znění pozdějších revizí. V případě, že se jedná o „ověřování nových postupů použitím metody, která dosud nebyla v klinické praxi na živém člověku zavedena“, postupuje se podle zákona č. 373/2011 Sb., o specifických zdravotních službách, ve znění pozdějších předpisů. Takové ověřování nové metody musí být provázeno informovaným souhlasem pacienta, schválením Vědeckou radou a Etickou komisí FN Motol. Ověřování

nezavedené metody taktéž vyžaduje povolení Ministerstva zdravotnictví poskytovateli (FN Motol) – zajišťuje Úsek pro vědu a výzkum, eventuelně Ústavu pro jadernou bezpečnost, pokud je metoda spojena s lékařským ozářením. Takové ověřování probíhá za sledování Vědeckou radou FN Motol.

9.2. Jsou používány validované nebo verifikované laboratorní metody a jsou vypracovány závazné dokumenty v souladu s normou ČSN EN ISO 15189, případně s dalšími legislativními předpisy a doporučeními odborných společností. Všechny léčebné postupy s využitím známých či nových léků, buněk nebo tkání musí být podrobeny dle zákona kontrole SÚKLu. U všech laboratorních metod je vyžadována externí kontrola kvality, zajišťovaná buď českými či zahraničními agenturami.

10 Ostatní formy klinického výzkumu

10.1. Výzkum na zvířatech se ve FN Motol neprovádí. Zaměstnanci FN Motol se však mohou výzkumu na zvířatech účastnit v rámci spolupráce na výzkumných projektech organizovaných jinými subjekty, a to na základě ohlášení své účasti Vědecké radě FN Motol. V těchto případech se výzkum řídí pravidly hlavního řešitele.

10.2. Případné další možné specifické formy výzkumu posoudí individuálně Vědecká rada a Etická komise FN Motol, které zajistí kontrolu nad jejich prováděním.

11 Využití výsledků vědy a výzkumu

11.1 Pravidla využití získaných prostředků

11.1.1. Prostředky smluvně získané z nejrůznějších typů stávajících grantů jsou rámcově rozdělovány na investiční a neinvestiční prostředky. Investičními prostředky jsou financovány investice pro vědu a výzkum, neinvestiční prostředky jsou členěny na běžné finanční výdaje, režii a osobní finanční prostředky. Toto rozdělení je předem specifikováno v grantovém projektu, podle kterého se využití prostředků řídí. Osobní prostředky jsou rezervovány dle platných předpisů na zajištění částečných i plných pracovních úvazků s preferencí pro nové mladé perspektivní vědecko-výzkumné pracovníky pro stávající i nově vznikající obory. Režie projektů je stanovena dle současných platných předpisů poskytovatelů a umožní zajistit režijní podporu výzkumu v rámci technického a personálního vybavení schválených a obhájených projektů.

11.1.2. Sankce za porušení smlouvy jsou definovány v jednotlivých smlouvách. Jako základní preventivní opatření slouží pravidelné roční hodnocení dosažených výsledků charakterizovaných především recenzovanými/impaktovanými publikacemi na vědeckých konferencích a závěrech interního auditu specializovanými odborníky pro jednotlivé směry výzkumu v rámci FN Motol.

11.1.3. Dosažené výsledky vědy a výzkumu jsou důvěrné do doby, než jsou se souhlasem řešitelů publikovány v domácím či zahraničním tisku, nebo do doby, než je nová diagnostická či léčebná metoda schválena odbornou společností či příslušnými odbory MZ ČR, případně je předmětem procesu ochrany duševního vlastnictví (užitné vzory, patenty).

11.1.4. Příjemce FN Motol poskytuje dosažené výsledky za stejných podmínek všem zájemcům k využití formou publikací v odborných impaktovaných/recenzovaných časopisech domácích i zahraničních, ve sdělovacích prostředcích a populárně vědeckých knihách či časopisech.

11.2 Využití výsledků výzkumu a vývoje a úprava vlastnických práv

11.2.1. Klinické hodnocení léčiv, klinického hodnocení zdravotnických prostředků a klinické zkoušky zdravotnických prostředků

Výsledek klinického hodnocení léčiv zadané do FN Motol na základě smlouvy se zadavatelem je výlučně vlastnictvím zadavatele (upraveno ve smlouvě).

11.2.2. Účelová a institucionální podpora výzkumu

Výsledky výzkumu a vlastnická práva vyplývající z projektů účelové a institucionální podpory (granty AZV ČR, GAČR, TAČR, Institucionální podpora FN Motol), jsou oceněny a právně ošetřeny smluvně stanovenými podmínkami a specifickými předpisy jednotlivých poskytovatelů grantů v souladu s právními předpisy České republiky.

Smlouva o využití výsledků výzkumu a přístrojů pořízených v průběhu výzkumu za prostředky k tomu v grantových projektech určených je vždy uzavírána na základě zakotvení této povinnosti do všeobecných podmínek, v souladu s příslušnými ustanoveními zákona č. 194/2016 Sb. a zákona č. 130/2002 Sb. Při uzavírání smlouvy o využití výsledků se vychází z úpravy užívacích a vlastnických práv k výsledkům uvedené ve smlouvě o poskytnutí podpory uzavřené mezi poskytovatelem a příjemcem. Smlouvy jsou předávány Grantové agentuře. Do doby uzavření smlouvy o využití výsledků není příjemce oprávněn s výsledky řešení projektu disponovat (netýká se publikování v odborných časopisech).

11.2.3. Moderní terapie

Využití výsledků projektů Moderních terapií je přísně individuální a řeší se před započetím projektu a následně v jeho průběhu v souladu s povahou projektu. Při účasti více subjektů v daném projektu se vlastnická práva řeší smluvně v souladu s platnými právními předpisy České republiky. Případné patentové řízení či uplatnění užitného vzoru se opětovně řeší individuálně podle platných právních předpisů. Koordinuje jej Úsek pro vědu a výzkum ve spolupráci s odborným garantem - lékařem či vědeckým pracovníkem příslušného projektu.

11.2.4. Finanční podpora evropských a ostatních fondů (granty EU a FM EHP/Norsko)

Výsledky výzkumu a vývoje z finanční podpory evropských a ostatních fondů jsou využívány v souladu s pravidly stanovenými pro jednotlivé operační programy a projekty, platnými právními předpisy zejména Evropského společenství, zákonem č. 121/2000 Sb., autorský zákon, ve znění pozdějších předpisů a zákonem č. 130/2002 Sb., o podpoře výzkumu, experimentálního vývojem a inovací, ve znění pozdějších předpisů.

Výsledky tohoto výzkumu a vývoje musí prokázat projekt hlavně v době udržitelnosti projektu. Pokud bude realizován zisk, musí být investován zpět do výzkumu a vývoje. Tento postup znovu investování bude popsán u každého projektu individuálně dle účelu projektu v souladu s právními předpisy týkajícími se využití výsledků výzkumu a vývoje.

11.2.5. Ostatní formy klinického výzkumu

Výsledky ostatních forem výzkumu zadávaných jinými organizacemi domácími či zahraničními podléhají smluvně určeným podmínkám mezi příjemcem a zadavatelem.

11.2.6. Nakládání s duševním vlastnictvím

FN Motol je oprávněna uzavírat licenční smlouvy – licence nevýhradní a úplatné. Dále je FN Motol oprávněna k převodům práv k duševnímu vlastnictví. Spoluvlastnické podíly by měly být určeny na základě míry přispění k dosažení jednotlivých výsledků, případně financí a měly by být podloženy znaleckým posudkem a smlouvou o určení spoluvlastnických podílů.

11.3 Využití dosažených výsledků k šíření nových poznatků pro veřejnost

11.3.1. Využití veřejných sdělovacích prostředků, populárně vědeckých časopisů, knih, informačních letáků, kampaní a internetových médií je důležitou součástí využití výzkumu, neboť umožňuje zdokonalit využití nových diagnostických, preventivních a léčebných metod, jejich kvalitní propagací u laické veřejnosti. Tím je znásoben sociální, ekonomický a zdravotnický dopad plánovaného výzkumu. U dotačních programů financovaných se spoluúčastí FN Motol je seznam publikačních povinností součástí realizačních podmínek s jejich splnění je jednou z podmínek pro udělení dotace. Návrh, provedení a organizace této popularizace výsledků výzkumu a vědy je v kompetenci Odboru komunikace s využitím obsahových podkladů poskytnutých řešitelem projektu. Informační akce, návštěvy a rozhovory s vybranými odborníky musí být vždy předem dohodnuty a odsouhlaseny nejenom v rámci klinik či ústavů FN Motol, ale také s vedením FN Motol a Odborem komunikace.

11.4 Využití výzkumem dosažených výsledků v pre- a postgraduální výuce

Tato aktivita je v rámci FN Motol řazena mezi zásadní povinnosti v souladu se strategií v oblasti vědy, výzkumu a inovací, řízených RVVI, neboť umožňuje využití nejnovějších výsledků domácího, ústavního výzkumu, ale i ověřování výsledků zahraničních v našich podmínkách a v naší populaci. Jde o určující strategii urychleného zavádění nových metod a postupů do léčebně-preventivní péče i podnětů k rozvoji další vědecko-výzkumné práce mezi perspektivními mladými vědeckými pracovníky ať už z lékařských fakult či přírodovědných, biochemických a biofyzikálních oborů.

11.5 Reinvestice získaných finančních a technických prostředků

Získané prostředky jsou zpětně investovány do dalšího aplikovaného výzkumu, experimentálního vývoje, zdokonalení technického vybavení pracovišť a do zdokonalení výuky a informačního šíření v odborné i laické veřejnosti dosažených výsledků. Dalším cílem využití zisku je zvyšování kvalifikace pracovníků jednotlivých klinických a laboratorních pracovišť, zabývajících se vědou a výzkumem ve Fakultní nemocnici v Motole.

11.6 Využití výsledků výzkumu v kategorii patentového řízení domácího či zahraničního

Využití takových výsledků se řeší individuálně podle podmínek zák. č. 527/1990 Sb., o vynálezech a zlepšovacích návrzích, ve znění pozdějších předpisů.

11.7 Doba využití výsledků

Doba využití výsledků je stanovena smluvně dle požadavků jednotlivých projektů, případně do doby využitelnosti těchto výsledků a v souladu s platnými právními předpisy.

12 Související předpisy

- Smlouva o Evropské unii a o fungování Evropské unie, Úřední věstník Evropské unie (2010/C 83/01), čl. 107
- Nařízení Komise (ES) č. 800/2008 ze dne 6.8.2008
- Nařízení Evropského parlamentu a Rady (EU) č. 536/2014 ze dne 16.dubna 2014 o klinických hodnoceních humánních léčivých přípravků a o zrušení směrnice 2001/20ES
- Nařízení Evropského parlamentu a Rady (EU) č. 2016/679 ze dne 27.dubna 2016 o ochraně fyzických osob v souvislosti se zpracováním osobních údajů a o volném pohybu těchto údajů a o zrušení směrnice 95/46/ES (obecné nařízení o ochraně osobních údajů)
- Rámec Společenství pro státní podporu výzkumu, experimentálního vývoje a inovací (uveřejněn v Úředním věstníku EU dne 30. 12. 2006/C323/01), článek 9
- Zákon č. 130/2002 Sb., o podpoře výzkumu, experimentálního vývoje a inovací z veřejných prostředků a o změně některých souvisejících zákonů (zákon o podpoře výzkumu, experimentálního vývoje a inovací), ve znění pozdějších předpisů
- Zákon č. 215/2004 Sb., o úpravě některých vztahů v oblasti veřejné podpory a o změně zákona o podpoře výzkumu a vývoje, ve znění pozdějších předpisů
- Zákon č. 527/1990 Sb., o vynálezech a zlepšovacích návrzích, ve znění pozdějších předpisů
- Zákon č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých souvisejících zákonů (autorský zákon), ve znění pozdějších předpisů
- Zákon č. 110/2019 Sb., o zpracování osobních údajů, ve znění pozdějších předpisů
- Zákon č. 268/2014 Sb., o zdravotnických prostředcích a o změně zákona č.634/2014 Sb. o správních poplatcích, ve znění pozdějších předpisů
- Zákon č. 378/2007 Sb., o léčivech a o změně některých souvisejících zákonů (zákon o léčivech), ve znění pozdějších předpisů
- Zákon č. 372/2011 Sb., o zdravotních službách a podmínkách jejich poskytování, ve znění pozdějších předpisů
- Zákon č. 373/2011 Sb., o specifických zdravotních službách, ve znění pozdějších předpisů
- Zákon č. 297/2016 Sb., o službách vytvářejících důvěru pro elektronické transakce
- Zákon č.218/2000 Sb., o rozpočtových pravidlech a o změně některých souvisejících zákonů (rozpočtová pravidla), ve znění pozdějších předpisů
- Zákon č. 527/1990 Sb., o vynálezech a zlepšovacích návrzích, ve znění pozdějších předpisů.
- Směrnice č. IOS_3/2004-4 Zavedení nového léčebného postupu, ve znění pozdějších revizí
- Směrnice č. IOS_10/2005-7 Informovaný souhlas pacienta, ve znění pozdějších revizí
- Směrnice č. IOS_3/2010-3 Klinické hodnocení, ve znění pozdějších revizí
- Směrnice č. IOS_7/2002-4 Interní granty FN Motol, ve znění pozdějších revizí
- Směrnice č. IOS_4/2018-1 Ochrana osobních údajů, ve znění pozdějších revizí

13 Závěrečná ustanovení

Tato verze směrnice č. IOS_1/2011-5 nabývá účinnosti dne 1. 10. 2020 a tímto dnem se ruší předchozí verze směrnice č. IOS_1/2011-4.

Zpracoval:

V Praze dne .....

.....

Prof. MUDr. Anna Šedivá, DSc.

Ověřil:

V Praze dne .....

.....

MUDr. Martin Holcát, MBA

Schválil:

V Praze dne .....

.....

JUDr. Ing. Miloslav Ludvík, MBA

ředitel nemocnice

Institutional Evaluation of the Predictive Module (Original Czech Text)

D

Nově implementovaný modul „Predikce rizik pomocí strojového učení“ v rámci Czech Salivary Gland Database představuje významné rozšíření funkčních možností systému směrem k pokročilé klinické podpoře rozhodování. Modul využívá kombinaci algoritmů Random Survival Forest a Cox Proportional Hazards Model pro modelování dvou klíčových cílových proměnných – celkového přežití a rizika recidivy. Trénování modelů probíhá na reálných, průběžně aktualizovaných datech pacientů uložených v databázi, což významně zvyšuje jejich externí validitu a klinickou relevanci. Na základě těchto modelů je možné u nově zařazených pacientů odhadnout individuální rizikový profil, který má potenciál přímé aplikace v prognostické stratifikaci, plánování dispenzarizace i volbě léčebné strategie. Zásadní výhodou systému je možnost opakovaného přetrénování modelů při doplňování nových dat nebo aktualizaci klinických outcome parametrů, což umožňuje kontinuální zlepšování predikční přesnosti. Tento dynamický přístup činí z daného modulu perspektivní nástroj pro implementaci principů personalizované medicíny v oblasti zhoubných nádorů slinných žláz.

Prognostic Module – User Guide

The prognostic module allows users to calculate risk predictions for individual patients using trained machine learning models, retrain models on the current database, and inspect information about all available models.

The complete Czech version of the manual is available in the `README.md` file on GitHub[1].

E.1 Calculating a Prediction

1. Navigate to the *Patient List* section.
2. Select the patient for whom you want to calculate a prognostic score.
3. Click the head-with-gear icon in the upper right corner (Figure E.1).

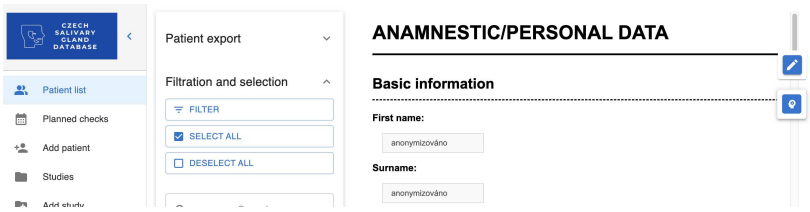


Figure E.1: Opening the prognostic module for a patient.

4. Select the type of prediction you want to calculate (Figure E.2).

[1] `https://github.com/vjelinekk/CzechSalivaryGlandDB`

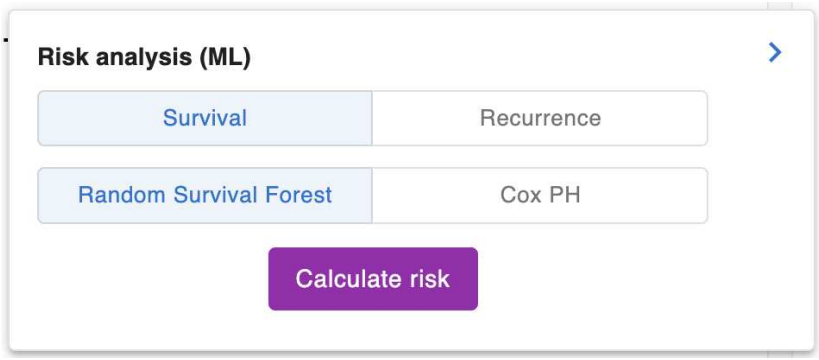


Figure E.2: Selecting the prediction type.

5. Click the **Calculate risk** button.
6. The calculation will run and the result will be displayed (Figure E.3).

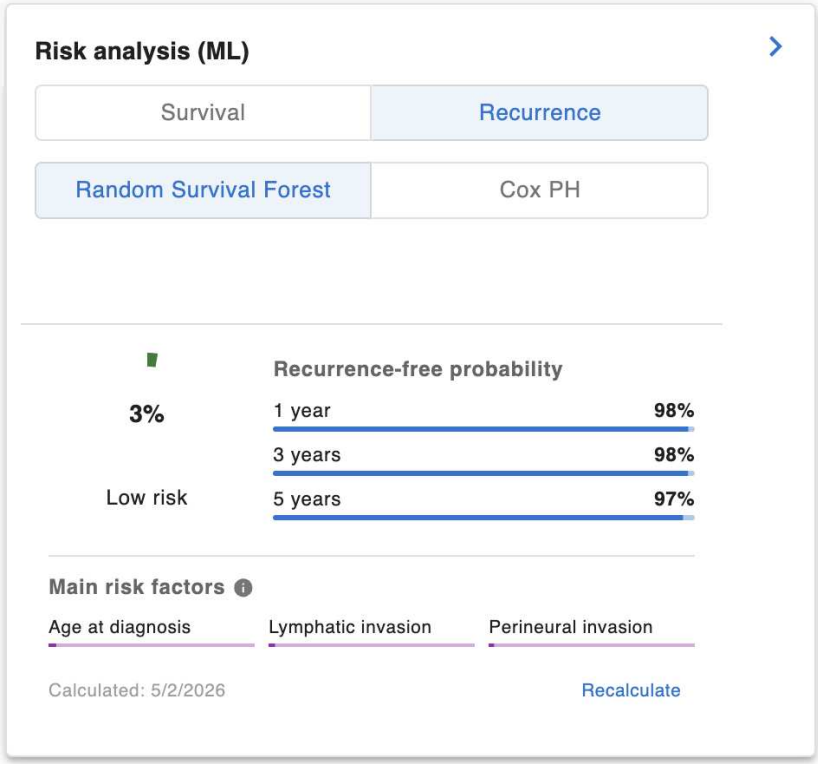


Figure E.3: Prediction result.

E.2 Training a Model

1. Navigate to the *Risk Prediction (ML)* section.

2. Select which model you want to train (Figure E.4).

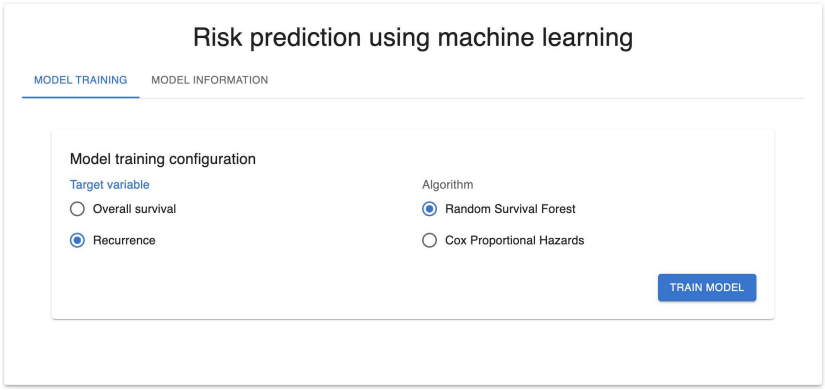


Figure E.4: Selecting a model for training.

3. Start the training by clicking the **TRAIN MODEL** button.

4. Once training is complete, the training result is displayed (Figure E.5).

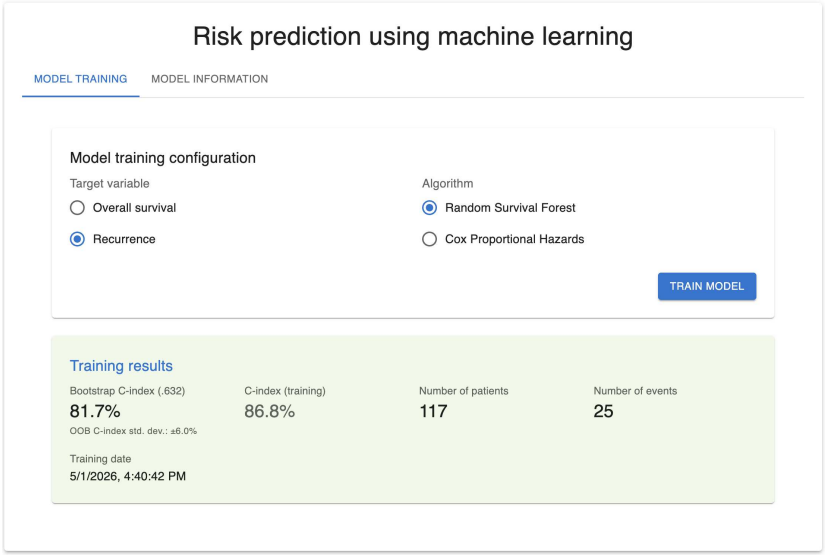


Figure E.5: Training result after completion.

E.3 Model Information

1. Navigate to the *Risk Prediction (ML)* section.

2. Switch to the *Model Information* tab (Figure E.6).

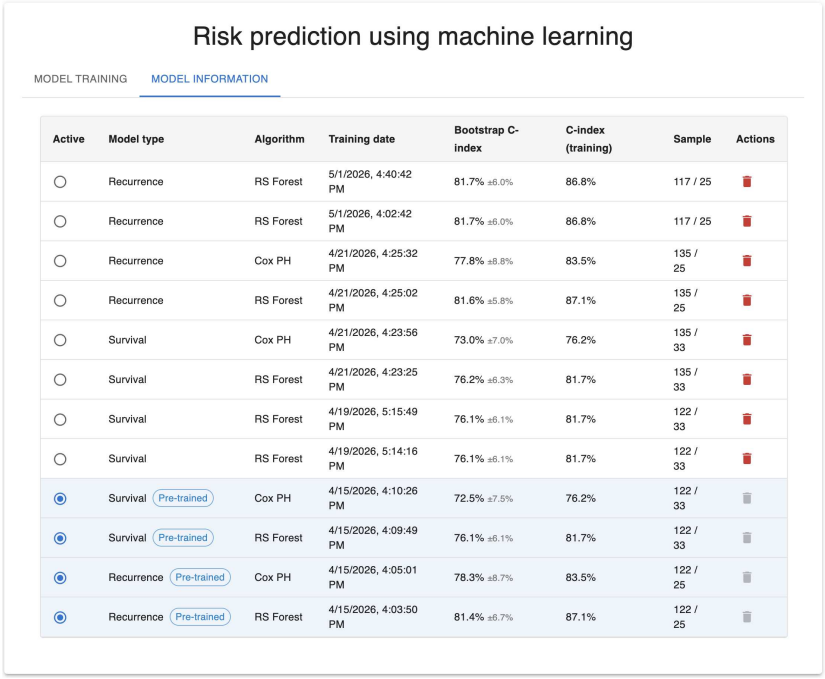


Figure E.6: Model information tab showing all trained and active models.

3. A list of all trained and currently active models is shown here.
4. You can select the active model and also delete not active models.

Attachment Contents F

The developed application and the text of this master's thesis are stored in the attached file `A24N0105P_prilohy.zip`. The individual directories are listed as headings in the following text.

F.1 Application_and_libraries

Source code of the Czech Salivary Gland Database desktop application (Electron, React, TypeScript, SQLite, Python ML sidecar).

- `src/` – application source split into three layers:
  - `main.ts` – Electron main process entry point
  - `preload.ts` – Electron preload script (context bridge)
  - `renderer.ts` – renderer process entry point
  - `root.tsx` – React root component
  - `index.html` – base HTML template
  - `backend/` – SQLite data layer and business logic (main process):
    - `db-entities/` – TypeScript interfaces mirroring each SQLite table
    - `mappers/` – conversion between raw DB entities and domain objects
    - `repositories/` – data-access layer wrapping SQL queries per domain area:
      - `patientRepository.ts` – CRUD and search operations for patients
      - `studyRepository.ts` – CRUD operations for research studies
      - `mlRepository.ts` – store and retrieve ML models and predictions

        - `passwordRepository.ts` – read/write the encrypted password hash
        - `tnmRepository.ts` – query TNM editions, rules, and value definitions
        - `dbHelpers.ts` – low-level query helpers shared across repositories
      - `services/` – business logic composed on top of repositories:
        - `patientService.ts` – patient creation, update, deletion, listing
        - `mlService.ts` – orchestrate train/predict calls to the Python sidecar and persist results
        - `exportService.ts` – build Excel/JSON exports from patient data
        - `importService.ts` – parse and validate import file contents
        - `importTransformService.ts` – transform raw import data into domain objects
        - `backupService.ts` – create and restore encrypted database backups
      - `utils/` – shared backend utilities:
        - `encryption.ts` – AES encryption/decryption helpers for database backups
        - `patientEncryption.ts` – field-level encryption for sensitive patient data
        - `mlManager.ts` – spawn, communicate with, and terminate the Python sidecar process
  - `frontend/` – React UI running in the renderer process:
    - `components/` – React components organised by feature area (patient list, forms, Kaplan–Meier chart, ML risk scoring, statistics, etc.)
    - `hooks/` – custom React hooks for form state, TNM computation, dispensarisation schedule, etc.
    - `utils/` – utility functions (Kaplan–Meier calculations, statistical tests, Czech PID validation, etc.)
    - `css/` – global and feature-specific stylesheets
  - `ipc/` – inter-process communication bridge (channel constants, IPC handler registrations for patients, ML, backup, export, import, encryption, and file-system operations)

- `python-ml-engine/` – Python sidecar process for survival analysis models (Cox PH, RSF) via scikit-survival, packaged as a standalone binary with PyInstaller:
  - `ml_engine.py` – main entry point (JSON stdin/stdout protocol)
  - `survival_model.py` – model training and prediction logic
  - `feature_extractor.py` – derives model features from raw patient data
  - `input_generator.py` – constructs model input structures
  - `validators.py` – input validation helpers
  - `ml_types/` – shared Python type definitions (enums, dataclasses for input, model, output, patient)
  - `test_ml_engine.py` – pytest tests for the engine entry point
  - `test_feature_extractor.py` – pytest tests for feature extraction
  - `test_validators.py` – pytest tests for validators
  - `requirements.txt` – Python dependencies
  - `ml_engine.spec` – PyInstaller spec file
  - `build.sh` – shell script to build the standalone binary
- `pretrained-models/` – pre-trained serialised model files (`.joblib`) for overall survival and recurrence, using Cox PH and RSF, shipped with the application
- `public/` – static assets: Arial fonts for PDF export, application icons and salivary-gland anatomical images, i18n JSON translation files (Czech, English, Slovak)
- `__tests__/` – Jest unit tests for React components and utility functions
- `package.json` – Node.js project manifest and npm scripts
- `tsconfig.json` – TypeScript compiler configuration
- `forge.config.ts` – Electron Forge packaging configuration
- `webpack.main.config.ts` – Webpack config for the main process
- `webpack.renderer.config.ts` – Webpack config for the renderer process
- `.github/workflows/` – GitHub Actions CI pipelines for build, test, and release

F.2 Input_data

Anonymised patient export files from the CSGDB database used as input for the model training experiments.

- `patients_k_13_3_26_priusni.xlsx` – parotid gland patient cohort
- `patients_k_13_3_26_podcelistni.xlsx` – submandibular gland patient cohort
- `patients_k_13_3_26_podjazykova.xlsx` – sublingual gland patient cohort

F.3 Results

Jupyter notebooks and supporting data for the survival analysis experiments underpinning the thesis validation chapter.

- `main.ipynb` – primary experiment notebook: all features, Cox PH and RSF models, C-index, Brier score, and feature importance
- `main_4_features.ipynb` – experiment restricted to the four most important features
- `main_4_features_improved.ipynb` – improved variant of the four-feature experiment with further hyperparameter tuning
- `COMPARISON.md` – written comparison and discussion of results across the three notebooks
- `README.md` – instructions for running the notebooks
- `requirements.txt` – Python dependencies for the notebook environment
- `data/patients.json` – patient dataset in JSON format used by the notebooks (derived from `Input_data/`)

F.4 Text_thesis

LaTeX source and compiled PDF of the Master's thesis.

- `main.tex` – single-file thesis source (all chapters, appendices, and bibliography)
- `main.pdf` – compiled thesis PDF

- `fasthesis.cls` – FASThesis LaTeX class file (v0.996) for UWB/FAV theses
- `acronyms.tex` – list of acronyms
- `cover.tex` – standalone cover page source
- `zadani.pdf` – official thesis assignment issued by the faculty
- `img/` – figures referenced from `main.tex`:
  - C4 architecture diagrams (Level 1–3, old and new CSGDB system)
  - ERA diagrams of the database schema (simplified, complete, and full-page)
  - Sidecar architectural pattern diagram
  - Model evaluation plots: integrated Brier score, feature importance
  - Application UI screenshots: patient risk card, ML risk scoring tab, prediction and training workflow steps, model information panel
  - Faculty and department logos, cover background graphics, QR code
  - Embedded directive PDF (IOS 1/2011-5)

F.5 Poster

Thesis poster.

- `poster.pdf` – final poster in PDF format
- `poster.pub` – editable poster source file (Microsoft Publisher)

List of Abbreviations

AdaBoost Adaptive Boosting

AFT Accelerated Failure Time

ANN Artificial Neural Networks

API Application Programming Interface

AUROC Area Under the Receiver Operating Characteristic

C-index Concordance Index

CDSS clinical decision support system

CHF Cumulative Hazard Function

CI Continuous Integration

Cox PH Cox Proportional Hazards

CSGDB Czech Salivary Gland Database

DAHANCA Danish Head and Neck Cancer Study Group

DTO Data Transfer Object

EMR Electronic Medical Record

ENE Extranodal Extension

EPV events per variable

ERA Entity-Relationship-Attribute

FNA Fine-Needle Aspiration

GBM Gradient Boosting Machines

GDPR General Data Protection Regulation

HIS Hospital Information System

HR Hazard Ratio

IBS Integrated Brier Score

ICCR International Collaboration on Cancer Reporting

IPC Inter-Process Communication

IPCW Inverse Probability of Censoring Weighting

LVI Lymphovascular Invasion

MLE Maximum Likelihood Estimation

MSE Mean Squared Error

MSRSGC Milan System for Reporting Salivary Gland Cytopathology

ND Neck Dissection

NN Neural Networks

OLS Ordinary Least Squares

OS Overall Survival

PH Proportional Hazards

PII Personal Identifiable Information

PNI Perineural Invasion

RF Random Forest

RFS Recurrence-Free Survival

ROM Risk of Malignancy

RSF Random Survival Forest

SEER Surveillance, Epidemiology, and End Results

SGC Salivary Gland Carcinoma

SHAP Shapley Additive exPlanations

TNM Tumor, Node, Metastasis

VIMP Variable Importance

Bibliography

1. HERPEN, Carla van et al. Salivary gland cancer: ESMO–European Reference Network on Rare Adult Solid Cancers (EURACAN) clinical practice guideline for diagnosis, treatment and follow-up. *ESMO open*. 2022, vol. 7, no. 6, p. 100602.
2. GUZZO, Marco et al. Major and minor salivary gland tumors. *Critical reviews in oncology/hematology*. 2010, vol. 74, no. 2, pp. 134–148.
3. SKÁLOVÁ, Alena; HYRCZA, Martin D; LEIVO, Ilmo. Update from the 5th edition of the World Health Organization classification of head and neck tumors: salivary glands. *Head and neck pathology*. 2022, vol. 16, no. 1, pp. 40–53.
4. SKALOVA, Alena; HYRCZA, Martin D. Proceedings of the North American society of head and neck pathology companion meeting, new Orleans, LA, March 12, 2023: classification of salivary gland tumors: Remaining controversial issues? *Head and Neck Pathology*. 2023, vol. 17, no. 2, pp. 285–291.
5. JANSEN, Louis et al. Incidence and survival of primary major salivary gland carcinoma in Germany over the last decade: A nationwide population-based study. *European Journal of Cancer*. 2025, p. 116196.
6. JELÍNEK, V. *Rozšíření možností Czech Salivary Gland Database pro klinickou praxi a pro analýzu dat*. 2024. Available also from: `https://theses.cz/id/4wvzj1/`. Bakalářská práce. Západočeská univerzita v Plzni. Supervisor: Ing. Petr Brůha.
7. JELÍNEK, Vojtěch; BRŮHA, Petr; KALFERT, David; NOVÝ, Pavel. Czech Salivary Gland Database. In: *Proceedings of the 18th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2025)*. SCITEPRESS, 2025, pp. 705–711. Available from DOI: `10.5220/0013254400003911`.
8. HANKEY, Benjamin F.; RIES, Lynn A.; EDWARDS, Brenda E. The Surveillance, Epidemiology, and End Results Program: A National Resource. *Cancer Epidemiology, Biomarkers & Prevention*. 1999, vol. 8, pp. 1117–1121.

9. FRIEDMAN, Samuel; NEGOITA, Sanda. History of the Surveillance, Epidemiology, and End Results (SEER) Program. *JNCI Monographs*. 2024, vol. 2024, no. 65, pp. 105–109. Available from DOI: 10.1093/jncimonographs/lgae033.

10. SEETHALA, Raja R. et al. *Carcinomas of the Major Salivary Glands Histopathology Reporting Guide*. 1st. International Collaboration on Cancer Reporting, 2018.

11. THOMPSON, Lester D. R. et al. *Carcinomas of the Major Salivary Glands Histopathology Reporting Guide*. 2nd. International Collaboration on Cancer Reporting, 2024.

12. OVERGAARD, Jens; JOVANOVIC, Aleksandar; GODBALLE, Christian; ERIKSEN, Jesper. The Danish Head and Neck Cancer database. *Clinical Epidemiology*. 2016, vol. 8, pp. 491–496. Available from DOI: 10.2147/CLEP.S103591.

13. KARTSONAKI, Christiana. Survival analysis. *Diagnostic Histopathology*. 2016, vol. 22, no. 7, pp. 263–270. ISSN 1756-2317. Available from DOI: 10.1016/j.mpdhp.2016.06.005. Mini-Symposium: Medical Statistics.

14. JENKINS, Stephen P. Survival analysis. *Unpublished manuscript, Institute for Social and Economic Research, University of Essex, Colchester, UK*. 2005, vol. 42, no. 54-56, p. 1.

15. MILLER JR, Rupert G. *Survival analysis*. John Wiley & Sons, 2011.

16. TURKSON, Anthony Joe; AYIAH-MENSAH, Francis; NIMOH, Vivian. Handling censoring and censored data in survival analysis: a standalone systematic literature review. *International journal of mathematics and mathematical sciences*. 2021, vol. 2021, no. 1, p. 9307475.

17. KLEIN, John P; MOESCHBERGER, Melvin L. *Survival analysis: techniques for censored and truncated data*. Vol. 1230. Springer, 2003.

18. RASHEED, Mohammed. Analyzing applications and properties of the exponential continuous distribution in reliability and survival analysis. *Journal of Positive Sciences*. 2023, vol. 4, no. 5, pp. 71–79.

19. LEE, Elisa T; WANG, John. *Statistical methods for survival data analysis*. Vol. 476. John Wiley & Sons, 2003.

20. SAPHNER, Thomas; TORMEY, Douglass C; GRAY, Robert. Annual hazard rates of recurrence for breast cancer after primary therapy. *Journal of clinical oncology*. 1996, vol. 14, no. 10, pp. 2738–2746.

21. CARROLL, Kevin J. On the use and utility of the Weibull model in the analysis of survival data. *Controlled clinical trials*. 2003, vol. 24, no. 6, pp. 682–701.

22. COLLETT, David. *Modelling survival data in medical research*. Chapman and Hall/CRC, 2023.

23. MYUNG, In Jae. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*. 2003, vol. 47, no. 1, pp. 90–100.

24. TSIATIS, Anastasios A. *Semiparametric theory and missing data*. Springer, 2006.

25. COX, David R. Regression models and life-tables. *Journal of the royal statistical society: Series B (methodological)*. 1972, vol. 34, no. 2, pp. 187–202.

26. SOBIN, Leslie H; GOSPODAROWICZ, Mary K; WITTEKIND, Christian. *TNM classification of malignant tumours*. John Wiley & Sons, 2011.

27. GANLY, Ian et al. *AJCC Cancer Staging System: Salivary Glands: Version 9 of the AJCC Cancer Staging System*. Independently published, 2025. ISBN 9798275082845.

28. BALOCH, Zubair; FIELD, Andrew S; KATABI, Nora; WENIG, Bruce M. The Milan system for reporting salivary gland cytopathology. In: *The milan system for reporting salivary gland cytopathology*. Springer, 2018, pp. 1–9.

29. ZHOU, Zhi-Hua. *Machine learning*. Springer nature, 2021.

30. ISHWARAN, Hemant; KOGALUR, Udaya B; BLACKSTONE, Eugene H; LAUER, Michael S. Random survival forests. 2008.

31. FREUND, Yoav; SCHAPIRE, Robert E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*. 1997, vol. 55, no. 1, pp. 119–139.

32. FRIEDMAN, Jerome H. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*. 2001, vol. 29, no. 5, pp. 1189–1232. Available from DOI: 10.1214/aos/1013203451.

33. NATEKIN, Alexey; KNOLL, Alois. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*. 2013, vol. 7, p. 63623.

34. FARAGGI, David; SIMON, Richard. A neural network model for survival data. *Statistics in medicine*. 1995, vol. 14, no. 1, pp. 73–82.

35. KATZMAN, Jared L et al. DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC medical research methodology*. 2018, vol. 18, no. 1, p. 24.

36. HARRELL, Frank E; CALIFF, Robert M; PRYOR, David B; LEE, Kerry L; ROSATI, Robert A. Evaluating the yield of medical tests. *Jama*. 1982, vol. 247, no. 18, pp. 2543–2546.

37. LONGATO, Enrico; VETTORETTI, Martina; DI CAMILLO, Barbara. A practical perspective on the concordance index for the evaluation and selection of prognostic time-to-event models. *Journal of biomedical informatics*. 2020, vol. 108, p. 103496.

38. VAN CALSTER, Ben; MCLERNON, David J; VAN SMEDEN, Maarten; WYNANTS, Laure; STEYERBERG, Ewout W. Calibration: the Achilles heel of predictive analytics. *BMC medicine*. 2019, vol. 17, no. 1, p. 230.

39. GRAF, Erika; SCHMOOR, Claudia; SAUERBREI, Willi; SCHUMACHER, Martin. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in medicine*. 1999, vol. 18, no. 17-18, pp. 2529–2545.

40. EFRON, Bradley. Estimating the error rate of a prediction rule: improvement on cross-validation. *Journal of the American statistical association*. 1983, vol. 78, no. 382, pp. 316–331.

41. SCHOENFELD, David. Partial residuals for the proportional hazards regression model. *Biometrika*. 1982, vol. 69, no. 1, pp. 239–241.

42. VÁZQUEZ-INGELMO, Andrea; GARCÍA-HOLGADO, Alicia; GARCÍA-PEÑALVO, Francisco J. C4 model in a Software Engineering subject to ease the comprehension of UML and the software. In: *2020 IEEE Global Engineering Education Conference (EDUCON)*. 2020, pp. 919–924. Available from DOI: 10.1109/EDUCON45650.2020.9125335.

43. OPENJS FOUNDATION. *Process Model | Electron* [online]. ElectronJS, 2026. [visited on 2026-03-20]. Available from: https://www.electronjs.org/docs/latest/tutorial/process-model.

44. EUROPEAN PARLIAMENT AND COUNCIL. *Regulation (EU) 2016/679 of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation)*. 2016. Available also from: http://data.europa.eu/eli/reg/2016/679/oj.

45. ČESKÁ REPUBLIKA. *Zákon č. 110/2019 Sb., o zpracování osobních údajů ve znění zákona č. 448/2024 Sb., zákona č. 218/2025 Sb. a zákona č. 230/2025 Sb.* Sbírka zákonů ČR, 2019. Available also from: https://e-sbirka.gov.cz/sb/2019/110/2025-08-01. ELI: https://e-sbirka.gov.cz/eli/cz/sb/2019/110/2025-08-01.

46. FAKULTNÍ NEMOCNICE V MOTOLE. *Organizační směrnice č. IOS_1/2011-5: Klinický výzkum*. 2020. Verze 5, účinnost od 1. 10. 2020.

47. FAKULTNÍ NEMOCNICE V MOTOLE. *Souhlas se zpracováním osobních údajů: Vedení databáze pacientů v rámci studie*. 2024. Interní formulář pro informovaný souhlas.

48. PEDUZZI, Peter; CONCATO, John; FEINSTEIN, Alvan R; HOLFORD, Theodore R. Importance of events per independent variable in proportional hazards regression analysis II. Accuracy and precision of regression estimates. *Journal of clinical epidemiology*. 1995, vol. 48, no. 12, pp. 1503–1510.

49. VANDER POORTEN, Vincent LM et al. The development of a prognostic score for patients with parotid carcinoma. *Cancer: Interdisciplinary International Journal of the American Cancer Society*. 1999, vol. 85, no. 9, pp. 2057–2067.

50. WESTERGAARD-NIELSEN, Marie et al. Prognostic scoring models in parotid gland carcinoma. *Head & Neck*. 2021, vol. 43, no. 7, pp. 2081–2090.

51. PEEPERKORN, Sam et al. Validated prognostic nomograms for patients with parotid carcinoma predicting 2-and 5-year tumor recurrence-free interval probability. *Frontiers in Oncology*. 2020, vol. 10, p. 1535.

52. CHEN, Yufan et al. Prognostic risk factor of major salivary gland carcinomas and survival prediction model based on random survival forests. *Cancer Medicine*. 2023, vol. 12, no. 9, pp. 10899–10907.

53. INTERNATIONAL AGENCY FOR RESEARCH ON CANCER. *Czechia Fact Sheet: Cancer Today (Globocan 2022)*. 2022. Tech. rep. World Health Organization. Available also from: `https://gco.iarc.who.int/media/globocan/factsheets/populations/203-czechia-fact-sheet.pdf`. Accessed: 2026-04-28.

54. BEJANI, Mohammad Mahdi; GHATEE, Mehdi. A systematic review on overfitting control in shallow and deep neural networks. *Artificial Intelligence Review*. 2021, vol. 54, no. 8, pp. 6391–6438.

55. NOHARA, Yasunobu; MATSUMOTO, Koutarou; SOEJIMA, Hidehisa; NAKASHIMA, Naoki. Explanation of machine learning models using shapley additive explanation and application for real data in hospital. *Computer Methods and Programs in Biomedicine*. 2022, vol. 214, p. 106584.

56. PHILLIPS, P Jonathon et al. Four principles of explainable artificial intelligence. 2021.

57. THERNEAU, Terry M. *A Package for Survival Analysis in R*. 2021. Available also from: `https://CRAN.R-project.org/package=survival`. R package version 3.2-13.

58. ISHWARAN, Hemant; KOGALUR, Udaya B. *Random Forests for Survival, Regression, and Classification (RF-SRC)*. 2021. Available also from: `https://CRAN.R-project.org/package=randomForestSRC`. R package version 2.11.0.

59. PÖLSTERL, Sebastian. scikit-survival: A Library for Time-to-Event Analysis Built on Top of scikit-learn. *Journal of Machine Learning Research*. 2020, vol. 21, no. 212, pp. 1–6. Available also from: `http://jmlr.org/papers/v21/20-729.html`.

60. DAVIDSON-PILON, Cameron. lifelines: survival analysis in Python. *Journal of Open Source Software*. 2019, vol. 4, no. 40, p. 1317. Available from DOI: `10.21105/joss.01317`.

61. HOCWALD, Eitan et al. Prognostic factors in major salivary gland cancer. *The Laryngoscope*. 2001, vol. 111, no. 8, pp. 1434–1439.

62. STEYERBERG, Ewout W et al. *Clinical prediction models*. Vol. 201. Springer, 2019. No. 9.

63. OSHIRO, Thais Mayumi; PEREZ, Pedro Santoro; BARANAUSKAS, José Augusto. How many trees in a random forest? In: *International workshop on machine learning and data mining in pattern recognition*. Springer, 2012, pp. 154–168.

64. PAVLOU, Menelaos; AMBLER, Gareth; SEAMAN, Shaun; DE IORIO, Maria; OMAR, Rumana Z. Review and evaluation of penalised regression methods for risk prediction in low-dimensional data with few events. *Statistics in medicine*. 2016, vol. 35, no. 7, pp. 1159–1177.

65. SIMON, Noah; FRIEDMAN, Jerome H; HASTIE, Trevor; TIBSHIRANI, Rob. Regularization paths for Cox's proportional hazards model via coordinate descent. *Journal of statistical software*. 2011, vol. 39, pp. 1–13.

List of Figures

List of Tables

List of Listings