Chemotherapy optimization and patient model parameter estimation based on noisy measurements

Borbála Gergics[1,2], Melánia Puskás[1,2,*], Lilla Kisbenedek[1], Martin Ferenc Dömény[1], Levente Kovács[1], and Dániel András Drexler[1]

[1]Physiological Controls Research Center, University Research and Innovation Center, Óbuda University, Budapest, Hungary.
[2]Applied Informatics and Applied Mathematics Doctoral School, Óbuda University, Budapest, Hungary.
[*]Corresponding: puskas.melania@uni-obuda.hu

Abstract:

The application of the achievements of mathematics and informatics greatly helped the development of medicine. Designing personalized therapies using different algorithms is crucial, especially during chemotherapy, to minimize the toxic effects on the patient and avoid resistance, thus ensuring a higher quality of life. In this work, we present an LSTM neural network that can quickly and accurately estimate the parameters of the tumor dynamics model based on noisy virtual patient data. In addition, we present a genetic algorithm designed for therapy optimization, which is able to predict the most appropriate personalized therapy based on the estimated parameters. In this work, we focus on finding the optimal hyperparameters of this genetic algorithm. Optimizing the hyperparameters is of fundamental importance in designing the best possible personalized therapy.

Keywords: LSTM recurrent neural network; genetic algorithm; therapy optimization; noise model; parameter estimation

1 Introduction

Cancer is a highly complex disease with different symptoms based on the location of the tumor and its progression. Besides the malignant tumor cells, the tumor consists of several other cell types and biological factors such as immune cells, cancer-associated fibroblasts, endothelial cells, pericytes, and various additional tissue-resident cell types. This heterogeneous system, together with the characteristics of the patients, leads to difficulties in cancer treatment [1–3].

With the development of science, several strategies were invented to cure cancer, for instance, radiotherapy, chemotherapy, surgical excision, immunotherapy, or the Rotational Field Quantum Magnetic Resonance (RFQMR) method. The main problem with these methods is that they are designed for the average patient and several are not available because of economic reasons. Inventing new curing methods could be money- and time-consuming as well, thus developing these well-known therapies is crucial to achieve more effective optimal treatments [4].

Informatics and mathematics in the last decades have been widely used in cancer treatment optimization. The dosing regimens and the optimal therapy can be examined with cytotoxicity models that describe the effect of chemotherapeutics on proliferation, cell signaling models pinpointing transition rates for drug targeting, and tissue scale models predicting tumor responses based on patient-specific imaging data [5]. Mainly, numerous researchers create mathematical models to describe the processes of pharmacokinetics and pharmacodynamics [6]. Subsequently, these researchers employ computational techniques to solve these models and determine the optimal chemotherapy. This plan typically outlines the specific combination, frequency, and dosage of drug administration [7]. Various algorithms are designed to determine the parameters of these models based on the characteristics of the patients. With the appropriate parameters, the clinical outcome can be predicted and the optimal therapy is definable.

Parameter identification problems can be established with neural networks faster than with the traditionally used differential equation solvers [8–10]. Here we carry out parameter estimation using Long Short-Term Memory (LSTM) neural networks. LSTM is designed to overcome the vanishing gradient problem of traditional recurrent neural networks (RNNs) which makes its architecture preferable for parameter estimation [11]. However, the usage of LSTM for parameter estimation related to tumor dynamics is rarely found in the literature. In Section 4 we propose a novel framework of parameter estimation of the tumor dynamics model detailed in Section 3.1. For network training purposes, we generated noisy measurement values. Due to the small amount of available mouse experiment data and the inaccuracy of measuring tumor volume, modeling the noise is necessary while simulating virtual patients. Subsection 3.2 discusses the method of noise modeling.

By selecting the correct parameters, it becomes possible to solve the optimization problem of personalized therapy. In this work, we focus on genetic algorithms utilized for therapy optimization. The genetic algorithm proposed in this work (Section 5) is based on metronomic chemotherapy principles [12]. Here, we implement the determination of the optimal hyperparameter of the genetic algorithm to achieve the most appropriate chemotherapy.

2 Related Works

Long Short-Term Memory (LSTM) networks are often preferred over simple feedforward neural networks for parameter estimation of dynamical systems due to their ability to capture and model temporal dependencies, which are crucial in dynamic systems. In these systems, the current state often depends on past states. Also, LSTM networks tend to be more robust on noisy and incomplete data. They can learn to filter out irrelevant information and focus on the most relevant aspects of the input sequences. Moreover, they can handle different lengths of time series inputs, due to the flexible architecture, while the traditional feed-forward networks typically require fixed-size input vectors [13, 14]. LSTM is broadly used in cancer research, Wu et al. implemented LSTM models to predict cancer risk when trained using a large but incomplete real-world dataset of tumor marker values. [15]. In the work of Wang et al., LSTM was used to predict respiration motion during real-time tracking of thoracic-abdominal tumors [16].

LSTM networks have been employed to address the inverse problem posed by differential equations and have found application in parameter estimation across various fields [17]. Zou et al. proposed an LSTM neural network to estimate the parameters of the pharmacokinetics of microvasculature and perfusion in normal and diseased tissues [18]. However, only a limited number of studies have utilized the LSTM approach for parameter identification in the context of tumor growth models [19]. In the work of Guo et al. LSTM was designed to overcome the inverse parameter estimation problem of partial differential equations generating simulated growth of low-grade gliomas with different clinical parameters [20].

Our preliminary works consist of parameter estimation of real measurement data based on mouse experiments employing neural networks trained on *in silico* virtual patient measurements. This preliminary study results suggest that the established model is applicable as an unconstrained optimization method for parameter fitting [21]. In other preliminaries, we introduced a method to identify potential parameter ranges and initial parameter values by generating an artificial time series of tumor volumes in a simulated environment. Then, Self-Organizing Maps were applied to categorize these time series into distinct clusters. The corresponding associated parameters were determined by identifying clusters exhibiting similar tumor dynamics [22].

In this work, we aim to create an LSTM network to estimate the parameters of the mathematical model describing tumor dynamics [23]. We trained the neural network with previously generated noisy measurement values. During preclinical experiments, tumor volume measurements are usually carried out with a digital caliper, which causes inaccuracy in volume values. Modeling this measurement noise is significant while generating *in silico* dataset [24], and also during more precise parameter estimation [21, 25–29] and therapy optimization [30–34].

Once we estimate the model parameters based on the noisy measurement data of the patients, the optimal personalized chemotherapy can be designed. This search or optimization problem has a fairly large search space and many independent variables. To solve such a problem, there is a wide variety of algorithms we can use [35]. Here, we achieve this with the help of genetic algorithms. The genetic algorithm is a global optimization technique frequently used in engineering problems due to its robustness and flexibility. The genetic algorithm aims to mimic the Darwinian theory of evolution happening in nature.

The application of genetic algorithms to therapy optimization is extensive [36]. Yu et al. achieved successful optimization in the design of stereotactic radiosurgery and radiotherapy. A self-contained system was introduced that integrates decision-theoretic guidance with a genetic algorithm to achieve optimization [37]. As early detection of cancer improves the outcome of therapy, applying genetic algorithm-partial least square-discriminant analysis system to analyze Raman spectroscopy to differentiate between a normal and dysplastic cervix offers valuable results conducted by Duraipandian et al. [38]. Predicting the prognosis of cancer offers informative factors affecting the appropriate therapy. Bozcuk et al. proposed a genetic algorithm-based method for comparison of different data mining methods to predict the outcome of cancer patients after hospitalization [39]. In our previous work, we have shown that the algorithm is capable of optimizing therapy for individual patients, as well as a whole group of patients with similar parameters [33, 40]. The genetic algorithm proposed in this work aims to create chemotherapies using the smallest dose possible to avoid the maximum tolerated doses (MTD) method.

3 Modeling tumor growth and generating noisy measurements for training data

3.1 Mathematical model of tumor growth

In order to generate training data for the LSTM network, we need to simulate tumor growth [28]. Tumor volume measurements make up the largest part of the training data. Since there are few tumor measurements from real experiments available for training the network, we have to multiply the number of training samples based on real measurements. For the simulation, we use a model described by a system of differential equations with four state variables, which models the dynamics of the tumor and the drug [41]:

$$\begin{aligned}
\dot{x}_1 &= (a-n)x_1 - b\frac{x_1x_3}{ED_{50}+x_3}, && (1)\\
\dot{x}_2 &= nx_1 + b\frac{x_1x_3}{ED_{50}+x_3} - wx_2, && (2)\\
\dot{x}_3 &= -(c+k_1)x_3 + k_2x_4 + u, && (3)\\
\dot{x}_4 &= k_1x_3 - k_2x_4, && (4)
\end{aligned}$$

where the state variables represent different time functions. The time function of the live tumor volume is denoted by x_1 [mm^3], and the time function of the dead tumor volume by x_2 [mm^3]. The flow of the drug between blood and tissues can be described with a two-compartment model, so x_3 [mg·kg^{-1}] is the time function of the drug level in the central compartment (blood), and x_4 [mg·kg^{-1}] is the time function of the drug level in the peripheral compartment (tissues).

The input of the system is the u [mg·kg^{-1}] injection rate. The input is impulsive since the change in the drug level due to the injection takes place in a very short time compared to the time constants of the system. Therefore, we can consider it as

	Parameter name	Unit	Min	Max
a	tumor growth rate coefficient	day^{-1}	0.24	0.68
b	drug efficiency rate coefficient	day^{-1}	4.88	30.00
n	necrosis rate coefficient	day^{-1}	$9.89 \cdot 10^{-3}$	$6.4 \cdot 10^{-2}$
w	washout rate coefficient of dead tumor cells	day^{-1}	$6.99 \cdot 10^{-2}$	0.10
ED_{50}	median effective dose of the drug	$\text{mg} \cdot \text{kg}^{-1}$	1.84	2.34
c	clearance of the drug	day^{-1}	1.82	1.82
k_1	flow rate coefficient of the drug from the central to the peripheral compartment	day^{-1}	14.01	14.01
k_2	flow rate coefficient of the drug from the peripheral to the central compartment	day^{-1}	136.28	136.28

Table 1
Name of the parameters of the model with units, minimum, and maximum value.

if there is a discontinuity in x_3 at time t_k, the size of which is d_k [23]:

$$x_3(t_k^+) = x_3(t_k^-) + d_k. \tag{5}$$

The output of the system is denoted by y, which is the volume of the total tumor, i.e., the sum of the living and dead tumor volume:

$$y = x_1 + x_2. \tag{6}$$

In reality, the tumor volume of the experimental mice is estimated using an approximate formula [42, 43]:

$$\text{V} = \frac{\pi}{2} (\text{width} \cdot \text{length})^{3/2}. \tag{7}$$

Since the measurements are performed with a digital caliper, we estimate the volume from the width and length of the tumor based on (7). The approximate formula assumes an ellipsoid shape for the tumor, which is not completely true in reality. The shape of the tumor is amorphous and, therefore, difficult to measure. Measurement noise is unavoidable when measurements are taken with a digital caliper and the approximation formula also contains measurement noise. The tumor can appear in hard-to-reach places on the body of animals, and the skin of animals often thickens during treatments [44].

Since real measurements are noisy, the LSTM is also trained on noisy measurements. In order to generate the most realistic training data, we use equations (1)-(4) to simulate the behavior of the tumor and generate noise for the resulting total tumor volumes (6) with the help of a noise model we prepared (detailed in Subsection 3.2).

The parameters of the model are summarized in Table 1. These parameters are unique for each patient. In our case, the estimation of these parameters determines how effective the individual therapy will be. We provide an initial estimate of the values of the parameters with the LSTM network. When generating the training data, we create the parameters using the minimum and maximum values determined by model fitting.

The pharmacokinetic parameters were acquired in a separate experiment specifically performed to investigate the drug pharmacokinetics. The same parameters c, k_1, and k_2 are used for all the patients, which were the results of the parametric identification carried out on the experimental data.

3.2 Generating noisy measurements for LSTM training

To generate the training data, we simulated the tumor volumes using (1)-(4), so we needed patient parameters for this simulation. These parameters were generated based on Table 1 using a normal distribution between the minimum and maximum values. Based on the current experimental protocol, experimental mice receive injections twice a week. Accordingly, one virtual mouse received an injection twice during the simulation. The doses were randomly generated between 0 mg/kg and 8 mg/kg according to the experimental protocol, also using a normal distribution.

During our experiments, the tumor is measured with a digital caliper, and the volume of the tumor is approximated based on the formula (7). However, we know that our measurements are noisy. If we know the measurement noise, it significantly helps the estimation of the parameters and makes neural networks more precise. Therefore, the modeling of measurement noise is essential in our research, since a more punctual knowledge of the parameters improves the optimization of the therapy [44].

The most accurate measurement tool is currently the pet MRI, but these measurements on small animals are complicated, as the animals must be anesthetized to use this tool. This is expensive and causes stress to the body of the animals, which also affects the experimental results, so in practice, the most common tool for measuring tumors is the digital caliper, which we use during our experiments. We have created a noise model that can model measurement noise in preclinical measurements, based on our current knowledge and data. The model was created based on MRI measurements from past cooperation and the corresponding digital caliper measurements [42]. The modeling process can be generalized even when MRI measurements are unavailable.

This noise model can be described based on Weibull distribution [44, 45] with a random variable x:

$$C(x,A,B,y,f,g) = \begin{cases} \frac{B}{A}\left(\frac{x}{A}\right)^{(B-1)} \exp\left(-\frac{x}{A}\right)^{B} \cdot \frac{1}{\sqrt[f]{\mathbf{y}+g}}, & \text{if } x \geq 0 \\ 0, & \text{if } x = 0 \end{cases} \tag{8}$$

We denoted this model with C. The parameters A and B are the scale and shape parameters of the Weibull distribution. The parameter y still represents the total tumor volume. The f and g are the parameters of a transformation that we used for noise whitening. The used noise whitening transformation is

$$\Phi_3(\mathbf{y}) = \sqrt[f]{\mathbf{y}+g}, \tag{9}$$

where $f = 1.0032$ and $g = 107.0034$. These values were obtained by parameter tuning. We needed the noise whitening because we fitted different probability distributions to the measurement noise. We fitted a total of 18 different probability distributions in our previous work [44] and tested the goodness of fit with the Anderson-Darling statistical test [46, 47]. However, the distributions did not take into account the fact that the measurement noise depended significantly on the size of the tumor being measured. We have already confirmed this finding in our previous work. Using the inverse of transformation (9) and the Weibull Distribution, the noise can be generated for the *in silico* measurements, which can be used to generate noisy measurements for neural network training.

The task of the neural network is to estimate the unique parameters of the patients based on noisy tumor volume measurements and the injected doses. A total of 100000 virtual mice tumor measurements and corresponding dose values were connected to the input of the network. The 100000 training samples were generated based on the real experimental protocol. Equations (1)-(4) and the parameters in Table 1 were used to simulate the total tumor volume. The parameters were randomized and generated with a normal distribution [48]. After that, the noise was generated for the total tumor volumes based on model (8). The doses were generated based on the normal distribution also. The composition of 10000 virtual mice consists of 100 different treatments (doses) of 1000 parameter sets (thus, we treated 1000 different mice in 100 ways).

Figure 1 illustrates a training sample. In the figure, blue circles indicate a simulated tumor volume and black circles indicate possible noisy points. When we trained the network, one measurement point had one noisy counterpart. The network was trained only from noisy measurement points and the doses. The figure illustrates what kind of noise can be generated for a simulated measurement using the noise model. The gray area shows the noise interval.

Figure 1 clearly shows that the generated noise is larger in the case of a small tumor volume and it is in a smaller interval in the case of a larger tumor volume. We have already proven this in our preliminary work [44].

4 Parameter estimation of the model with LSTM

Determining the parameters in the model (1)-(4) is crucial for designing optimal therapies. The parameters are usually found using search algorithms, and these processes often converge to local optima. One of our objectives is to broaden the search space by training a neural network capable of predicting initial parameter values closer to the actual ones. Additionally, this network can serve for standalone parameter identification purposes.

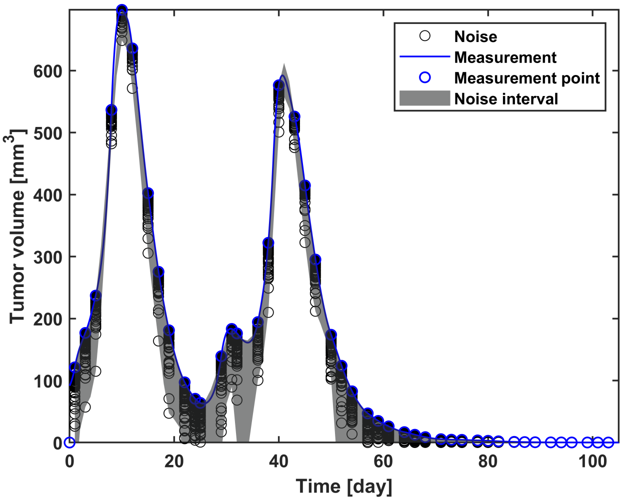

Figure 1
An example of a training sample (virtual patient), is where the simulated tumor volume measurements are indicated by the blue curve, the specified measurement points are shown by the blue circles and the noise generated for the measurement points is shown by the black circles. (The figure is illustrative, when the network was trained, one measuring point was accompanied by one noisy point.)

To train the neural network, *in silico* treatments are simulated using the tumor model. The generation of time series was done by solving numerically the differential equations as described in Subsection 3.2. The model inputs are the generated 100000 time series of the measurements of tumor volumes and the registered doses. The output of the model is the corresponding parameters.

Our aim is to predict the parameter values. One problem is that the available measurements are sparse, the tumor measurements can not be measured each day, so the network has to learn this sparsity. To simulate the sparsity, we removed the measurements on every even day and removed the injected drug values from every odd day. To overcome the sparsity problems, we used the *Masking* feature of the LSTM network. We used for training the first 5 days of the time series.

From the generated dataset, 100 data was eliminated, for final evaluation purposes. Then the remaining dataset is used for training the LSTM model. It was split into training and testing datasets with a 80 : 20 ratio. These are the $X \in \mathbb{R}^{N\times T\times F}$ inputs of the LSTM model, where N is the number of available time series. The T is the sequence length which is chosen to be 5, and F is the number of features, which is 2, because we used both the tumor volumes and the administered drug doses as inputs. The output of the model is $P \in \mathbb{R}^{N\times L}$, where L is 5, the number of identifiable parameters.

The LSTM recurrent neural network fundamental unit is the LSTM cell [49]. As the long short-term memory name implies, it can also use previously processed things for prediction. It consists of combinations of input-gate, memory-gate, and forget-gate. The forget gate, a key component, decides what information to discard from the cell state, allowing the model to selectively retain relevant information. The input gate updates the cell state with new information by determining what values to add. In our case, the objective was to enhance the retention of previously seen parameter values.

In the presented model, the sequences x_t with a length of T are processed consecutively, it is an LSTM unit input. An LSTM input consists of an input gate (i_t), a forget gate (f_t), and an output gate (o_t). One unit of an LSTM can be described as:

$$\begin{aligned}
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i), && (10)\\
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f), && (11)\\
g_t &= \tanh(W_g x_t + U_g h_{t-1} + b_g), && (12)\\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o), && (13)\\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t, && (14)\\
h_t &= o_t \odot \tanh(c_t), && (15)\\
& && (16)
\end{aligned}$$

where c_t is the cell state vector, while h_t is the output vector. W and U are weight matrices, while b is the bias vector. The network aims to adjust these matrices that way to learn to predict the output during training. The $\odot$ means Hadamard product (element-wise product) between matrices. The σ denotes the sigmoid function, while tanh is a hyperbolic tangent function [50].

The Keras Tensorflow API was employed to construct the network [51]. The network architecture comprises an input layer, a masking layer, and three LSTM layers utilizing the *Relu* activation function, each with 50 neurons. The output is a linear layer. *Nadam* optimizer was used during training when the learning rate was set to 0.01 and the batch size was adjusted to 64. The metric for evaluation was a mean squared error. During training, *EarlyStopping* method was used. *EarlyStopping* stops the training when the loss has not improved for a given number of iterations.

5 Therapy generation with genetic algorithm

If we have parameters used as virtual patient parameters (Section 3.2) and a mathematical model capable of describing tumor dynamics based on the injected doses (Section 3.1), we can generate an optimal therapy for the virtual patients.

As mentioned in Section 2, a previously created genetic algorithm can be used to generate optimal therapy for the virtual patient. This genetic algorithm treats the potential solutions to the given problem as the individuals in a virtual population and, through many generations, it ”breeds” an optimal solution using different genetic operators. In the case of chemotherapy optimization, the solutions to the problem are the treatments that can cure the patient, with as little injected doses as possible, to minimize harmful side effects and reduce the risk of developing drug resistance.

The fitness function determines the objective of the optimization, so it has to include the injected doses and the tumor volume. The used fitness function can be written as:

$$F = R_1 \sum_{i=1}^{N} \left(\frac{\varphi_i}{Y_{max}} \right)^2 + R_2 \sum_{i=1}^{N} \left(\frac{U_i}{U_{max}} \right)^2 + W_1 \sum_{i=1}^{N} U_i + W_2 Y_{end}, \tag{17}$$

where φ_i is given by:

$$\varphi_i = \begin{cases} Y_i - Y_{ref}, & \text{if } Y_i \geq Y_{ref}, \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

R_1, R_2, W_1 and W_2 represent weights, N is the number of genes each individual has and is equals to the length of the treatment. U_i is the injected dose on day i, and Y_i represents the tumor volume on day i. Y_{ref} is the reference tumor volume (the algorithm will attempt to shrink the tumor volume of the patients under this limit), and Y_{end} is the tumor volume at the end of the treatment. Lastly, U_{max} and Y_{max} are added to normalize the doses and the tumor values, respectively. The objective of the function is to minimize the injected doses and the tumor volumes at the same time. This value describes how effective the treatment is for the given patient [40].

The used selection operator is the tournament selection, which is proven to work well on this problem [33]. The selection operator is responsible for selecting the better solutions, discarding the rest, and creating the set of parents. This selection randomly selects individuals into smaller groups, and from each group, the fittest individual gets to inherit their genes:

$$\mathrm{P} = \arg\max_{E_i} F(E_i), \tag{19}$$

where P is the selected individual, E_i denotes the individuals where $E_i \in E$, where E is the set of the groups. The next step is the crossover function, which combines two randomly selected parents into a new individual. For this, we use the Laplace crossover [33, 40, 52]:

$$\mathrm{C}_i = \begin{cases} p_1 + \lambda |p_1 - p_2|, & \text{if } r < 0.5, \\ p_2 + \lambda |p_1 - p_2|, & \text{if } r \geq 0.5, \end{cases} \tag{20}$$

where C_i is the newly created individual, p_1 and p_2 are parents, the r parameter is a random number generated from uniform distribution and λ is a scaling factor generated randomly from Laplace distribution [52].

The last step is the mutation operator, which prevents the search from converging to local optima, by making random changes to the individuals. The used function is the Power mutation [40, 53]:

$$\mathrm{M}_i = \begin{cases} x_i - s(x_i - \alpha_i), & \text{if } t < r, \\ x_i + s(\beta_i - x_i), & \text{if } t \geq r, \end{cases} \tag{21}$$

where M_i is the mutated individual, x_i is the ith gene of the individual, α and β are the lower and upper bounds of the genes respectively, t is the scaled distance of x_i from the ith component of the lower bound (α_i) of the doses and s is given by [53]:

$$s = C g_i^{-\eta}, \tag{22}$$

where C is a constant scalar, η determines the slope and g_i is the current generation. This expression follows an exponential decay, initially having higher values, and as the generations progress, this value decreases exponentially [40].

Using these functions, we can generate a treatment, however, for the best results, we have to find the optimal hyperparameters of the genetic algorithm. Choosing the wrong parameters can increase the runtime of the algorithm significantly, or provide suboptimal solutions. The mutation rate is one of the most important hyperparameters, which determines a trade-off between local search and randomized global search.

6 Results

6.1 Accuracy of the parameter estimator network

To evaluate the LSTM prediction performance, we tested the model on 100, previously eliminated data from the training dataset. First, we calculated the relative estimation error compared to the true parameter values. In Figure 2, the relative estimation error of the five different parameters can be seen. The central rectangle, or box, represents the interquartile range (IQR), containing the middle 50% of the data. The lower bound (lower quartile or Q1) is the median of the lower half of the data set containing the errors, while the upper bound (upper quartile or Q3) is the median of the upper half. The green line marks the median of the given errors (Q2). These boxplots do not show the outliers, that are below and above the whiskers.

Since the first error metric did not provide representative information about model performance, another metric was calculated during the evaluation. In the new error, we divided the difference of parameters with the whole interval, that we used to create the training data. The interval limits are the maximum and the minimum value of the parameter in Table 1. In this case, the medians of the absolute values of the percentage errors are 7.68, 5.71, 12.97, 10.57, and 12.01, respectively. The model performance was the best in the case of b parameters. The real values are shown with the predicted values in Figure 3. The model is most accurate when the

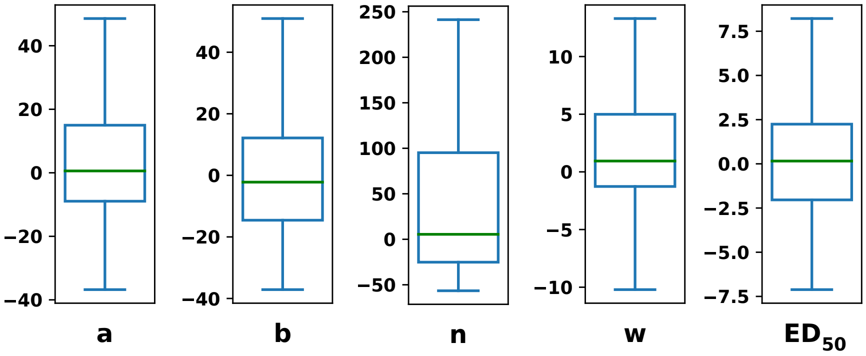

Figure 2
The relative estimation error of the parameters (a, b, n, w, ED_{50}) during evaluation. Each boxplot contains discrepancies in the case of 100 test parameters.

two values are the same, so the parameters are located on the 45-degree line.

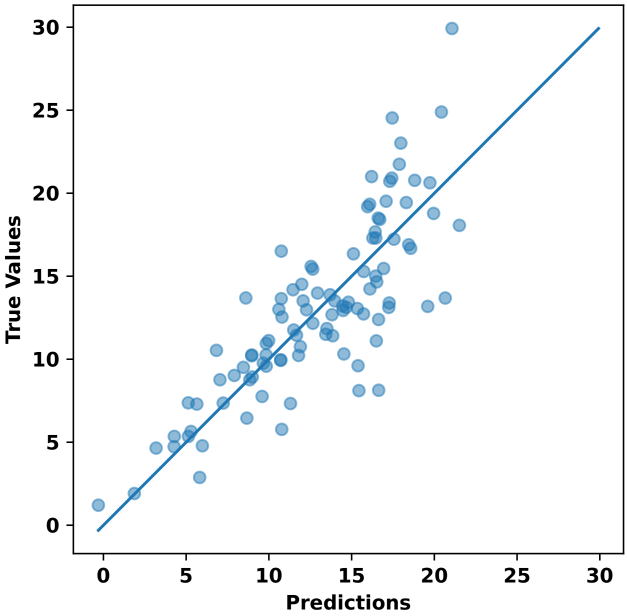

Figure 3
The scatter plot of 100 test cases in case of b parameter. The true values are visualized as a function of predicted parameters. The 45-degree line shows the punctual prediction.

6.2 Results of hyperparameter optimization and therapy generation

To determine the optimal hyperparameters, we used a method called grid search, which is simply an exhaustive search through a manually specified subset of the hyperparameter space [54]. We used the hyperparameters from our previous work as starting points and examined ranges of values around these points [33, 40]. Currently, the only stopping criteria for the algorithm is reaching the maximum number of generations. On all the results, an elite rate of 4% is used, and this value is not optimized.

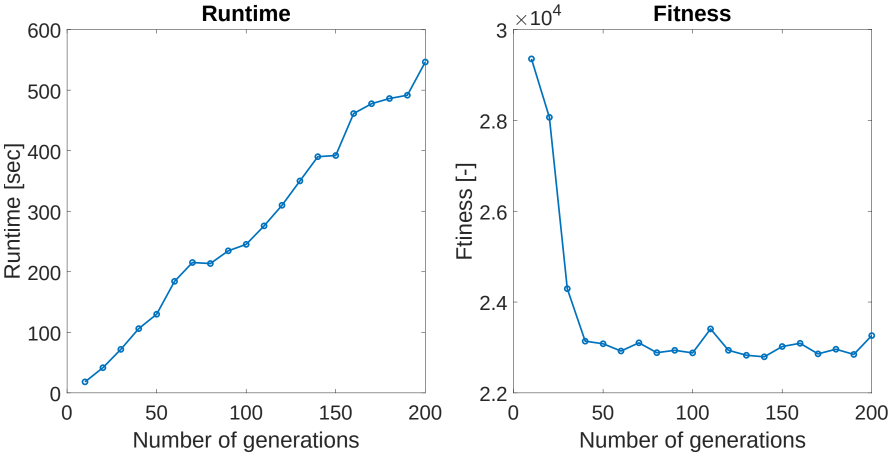

Figure 4
The change of runtime with different numbers of generations (left) and the change in fitness values with different numbers of generations (right). We can see a linear increase in runtime as the generations increase. The fitness value decreases as we increase the number of generations exponentially. In the figures, the mutation rate is 0.1 and the population size is 150.

In Figure 4, we can see the effects of the number of generations on the runtime and the fitness values. As we can see, the runtime has linear characteristics, increasing as the number of generations increases. The fitness value, however, decreases exponentially as the number of generations increases. After 80–100 generations, we only get a minimal decrease in fitness value at the price of a steady increase in runtime, so going higher is unnecessary.

In Figure 5 we can see the effects of the mutation rate. On the left, the runtime settles at a fairly constant value (around 250), with moderate fluctuations. On the right, we can see an exponential increase in fitness value with more noise, as we increase the mutation rate. This was expected, since with higher mutation, more random changes are applied to the population. The small fitness value at zero mutation rate suggests that the problem space does not have many local optima, since the algorithm could find a good solution without mutation. A mutation rate below 0.1 generally provided the best solutions.

In Figure 6 the runtime and fitness value at different population sizes is shown. The

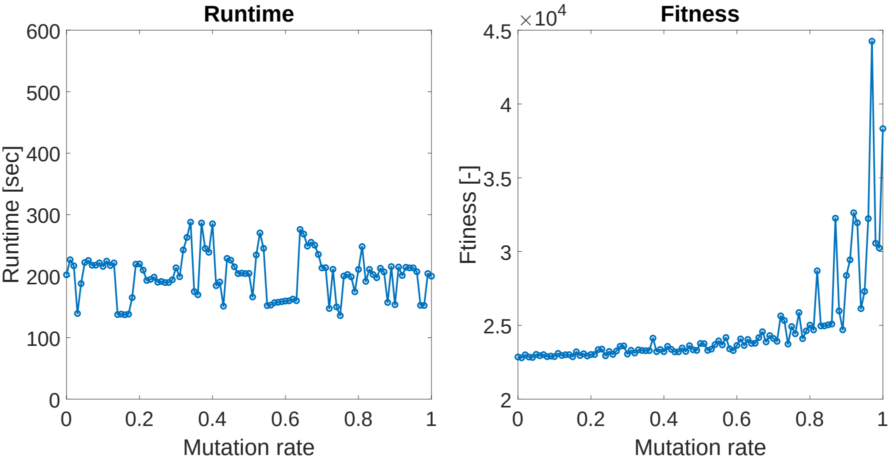

Figure 5

The change of runtime with different mutation rates (left) and the change in fitness values with different mutation rates (right). The runtime stays fairly constant with the change in mutation rate, however, the fitness value increases with the increase in the mutation rate. In the figures, the generation number is 80 and the population size is 150.

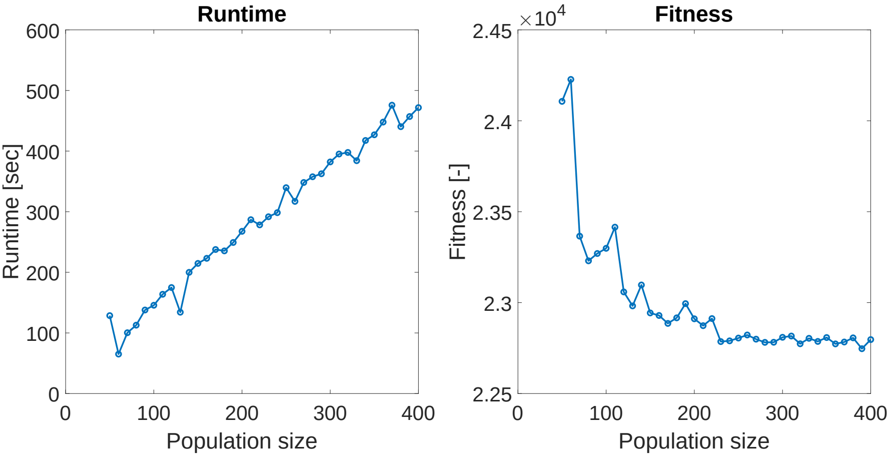

Figure 6

The change of runtime with population sizes (left) and the change in fitness values with different population sizes (right). Similarly to Figure 4, the runtime follows a linear increase, with the increase of the population size. The fitness value on the right follows an exponential decay, having a minimum value at around 300 population size. In the figures, the generation number is 80 and the mutation rate is 0.1.

runtime on the left increases linearly, just like in the previous case. The fitness value follows an exponential decay, reaching a fairly low value after 270-300 population size. In the literature, most researcher suggests that the population size should be dependent on the number of genes each individual has. In fact, a frequently used value for population size is 1.5-2 times the number of genes [55, 56]. This value is very close to our results but slightly lower. Of course, these values depend on the problem and the search space itself, so examining the optimal hyperparameter values is essential each time we apply a genetic algorithm to a new type of problem.

To test the parameters, we generated a therapy for a virtual patient. The result can be seen in Figure 7. The therapy was generated using 80 as the maximum number of generations, 0.1 as the mutation rate, and 300 as the population size. As we can see, the therapy successfully shrinks the tumor volume during the 105-day treatment.

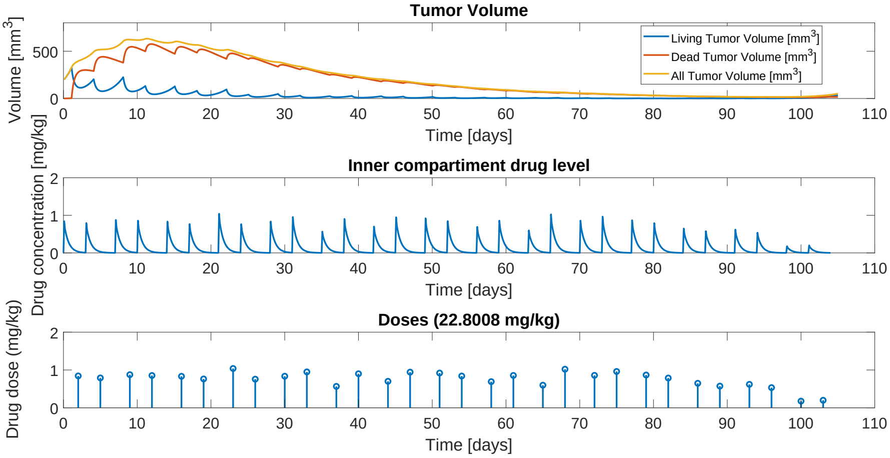

Figure 7
The result of a therapy optimization for a virtual patient. The top upper contains the living (blue), dead (red), and total (yellow) tumor volumes in time, the middle figure shows the concentration of the drug in the blood of the patient, and the bottom figure shows the injected doses on the given days.

7 Conclusion

In order to generate the *in silico* measurements for the neural network training, the real experimental setup must be modeled, where measurements were performed with a digital caliper (we can only measure the width and length of the tumor) and the tumor volume was estimated using the formula (7). However, the mathematical model (1)-(4) we use generates tumor volume values, so the deviations resulting from the measurement set-up are described with a noise model (8). The created noise model enables the generation of realistic *in silico* measurements. *In silico* measurements are essential for testing therapy and designing experiments and can be used to train neural networks.

The parameters were estimated using a neural network. The estimation of parameters is crucial in designing a unique personalized therapy. The more precisely the network estimates the parameters of the patient, the more effective therapy can be generated for the patient. In the case of parameter prediction, an LSTM recurrent neural network has been trained to identify the parameters of the tumor model. The trained model was the most punctual in the case of parameter b during evaluation.

Future plans include extending the input lengths and using the advantage of LSTM networks, where not only sparse data but different lengths of inputs can be fed. Moreover, another aim is to modify the architecture to use sequential inputs for training. A punctual parameter predictor network can reduce the time of parameter estimation. The trained LSTM neural network can quickly and accurately estimate the parameters of the tumor dynamics model. The neural network was trained using previously generated *in silico* measurement values with inherent noise.

In addition, we created a genetic algorithm specialized for therapy optimization. In this research, the emphasis was also on fine-tuning the hyperparameters of the genetic algorithm. Optimizing hyperparameters is key to developing the most effective personalized therapy. Based on the results, it can be concluded that genetic algorithm optimization is a promising direction for setting up the optimal treatment plan. Tuning the hyperparameters improves the effectiveness of the therapy. The genetic algorithm administers much lower doses than a standard clinical therapy, where patients receive maximally tolerated doses.

In summary, the noise model reproduces the measurement noise, which can be used for parameter identification and the generation of realistic, noisy measurement data. Neural networks are also able to learn from the simulated data of real noisy measurements and provide an appropriate estimate of the parameter values. In the future, networks are expected to be able to track changes in parameters and return parameter values from different measurement intervals. This makes it possible to generate a unique therapy. With the genetic algorithm, unique therapy can be generated using the estimated model parameters, and the development of resistance and toxicity can be avoided by the small dosage of chemotherapeutic drug. The result of our research could be a commercial medical device that can generate personalized therapy for cancer patients by estimating the unique parameters of the patient based on the measurements.

Acknowledgment

Project no. 2019-1.3.1-KK-2019-00007. has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the 2019-1.3.1-KK funding scheme. This research was supported partially by Horizon2020-2017-RISE-777911 project. This project has been supported by the Hungarian National Research, Development and Innovation Fund of Hungary, financed under the TKP2021-NKTA-36 funding scheme. The work of Dr. Dániel András Drexler was supported by the Starting Excellence Researcher Program of Óbuda University, Budapest, Hungary. Melánia Puskás was supported by the ÚNKP-23-3 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund. Lilla Kisbenedek was supported by the ÚNKP-23-2 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund. Martin Ferenc Dömény was supported by the ÚNKP-23-1 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.

References

[1] B. Arneth. Tumor microenvironment. *Medicina*, 56(1):15, 2019.

[2] T. Wu and Y. Dai. Tumor microenvironment and therapeutic response. *Cancer letters*, 387:61–68, 2017.

[3] M. Hellmann, B. Li, J. Chaft, and M. Kris. Chemotherapy remains an essential element of personalized care for persons with lung cancers. *Annals of Oncology*, 27(10):1829–1835, 2016.

[4] E. J. Mun, H. M. Babiker, U. Weinberg, E. D. Kirson, and D. D. Von Hoff. Tumor-treating fields: a fourth modality in cancer treatment. *Clinical Cancer Research*, 24(2):266–275, 2018.

[5] A. M. Jarrett, E. A. Lima, D. A. Hormuth, M. T. McKenna, X. Feng, D. A. Ekrut, A. C. M. Resende, A. Brock, and T. E. Yankeelov. Mathematical models of tumor cell proliferation: A review of the literature. *Expert review of anticancer therapy*, 18(12):1271–1286, 2018.

[6] J. M. Harrold and R. S. Parker. Clinically relevant cancer chemotherapy dose scheduling via mixed-integer optimization. *Computers & Chemical Engineering*, 33(12):2042–2054, 2009.

[7] H. Sbeity and R. Younes. Review of optimization methods for cancer chemotherapy treatment planning. *Journal of Computer Science & Systems Biology*, 8(2):74, 2015.

[8] M. Siket, G. Eigner, D. A. Drexler, I. Rudas, and L. Kovács. State and parameter estimation of a mathematical carcinoma model under chemotherapeutic treatment. *Applied Sciences*, 10(24):9046, 2020.

[9] T. Samad and A. Mathur. Parameter estimation for process control with neural networks. *International Journal of Approximate Reasoning*, 7(3-4):149–164, 1992.

[10] N. N. Son and L. T. Vinh. Parameter estimation of photovoltaic model, using balancing composite motion optimization. *ACTA POLYTECHNICA HUNGARICA*, 19(11):27–46, 2022.

[11] T. H. Kerbaa, A. Mezache, F. Gini, and M. S. Greco. Cnn-lstm based approach for parameter estimation of k-clutter plus noise. In *2020 IEEE Radar Conference (RadarConf20)*, pages 1–6. IEEE, 2020.

[12] I. Kareva, D. J. Waxman, and G. L. Klement. Metronomic chemotherapy: an attractive alternative to maximum tolerated dose therapy that can activate anti-tumor immunity and minimize therapeutic resistance. *Cancer letters*, 358(2):100–106, 2015.

[13] P. Malhotra, L. Vig, G. Shroff, P. Agarwal, et al. Long short term memory networks for anomaly detection in time series. In *Esann*, volume 2015, page 89, 2015.

[14] K. M. Tsiouris, V. C. Pezoulas, M. Zervakis, S. Konitsiotis, D. D. Koutsouris, and D. I. Fotiadis. A long short-term memory deep learning network for the

prediction of epileptic seizures using eeg signals. *Computers in biology and medicine*, 99:24–37, 2018.

[15] X. Wu, H.-Y. Wang, P. Shi, R. Sun, X. Wang, Z. Luo, F. Zeng, M. S. Lebowitz, W.-Y. Lin, J.-J. Lu, et al. Long short-term memory model–a deep learning approach for medical data with irregularity in cancer predication with tumor markers. *Computers in Biology and Medicine*, 144:105362, 2022.

[16] R. Wang, X. Liang, X. Zhu, and Y. Xie. A feasibility of respiration prediction based on deep bi-lstm for real-time tumor tracking. *IEEE Access*, 6:51262–51268, 2018.

[17] H. Chun, J. Kim, J. Yu, and S. Han. Real-time parameter estimation of an electrochemical lithium-ion battery model using a long short-term memory network. *IEEE Access*, 8:81789–81799, 2020.

[18] J. Zou, J. M. Balter, and Y. Cao. Estimation of pharmacokinetic parameters from dce-mri by extracting long and short time-dependent features using an lstm network. *Medical physics*, 47(8):3447–3457, 2020.

[19] M. Laurie and J. Lu. Explainable deep learning for tumor dynamic modeling and overall survival prediction using neural-ode. *npj Systems Biology and Applications*, 9(1):58, 2023.

[20] J. Guo, Z. Liang, G. Ditzler, N. C. Bouaynaya, E. Scribner, and H. M. Fathallah-Shaykh. Nonlinear brain tumor model estimation with long short-term memory neural networks. In *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, July 2018.

[21] L. Kisbenedek, M. Puskás, L. Kovács, and D. A. Drexler. Indirect supervised fine-tuning of a tumor model parameter estimator neural network. In *2023 IEEE 17th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pages 000109–000116, 2023.

[22] L. Kisbenedek, M. Puskás, L. Kovács, and D. A. Drexler. Clustering-based parameter estimation of a tumor model. In *IEEE 21st International Symposium on Intelligent Systems and Informatics (SISY 2023*, 2023.

[23] D. A. Drexler, T. Ferenci, A. Füredi, G. Szakács, and L. Kovács. Experimental data-driven tumor modeling for chemotherapy. In *Proceedings of the 21st IFAC World Congress*, pages 16466–16471, 2020.

[24] C. D. Man, F. Micheletto, D. Lv, M. Breton, B. Kovatchev, and C. Cobelli. The uva/padova type 1 diabetes simulator: New features. *Journal of diabetes science and technology*, 8(1):26–34, 2014.

[25] M. Puskás and D. A. Drexler. Parameter identification of a tumor model using artificial neural networks. In *Proceedings of the 2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics*, pages 443–448, 2021.

[26] M. Puskás and D. A. Drexler. Tumor model parameter estimation for therapy-optimization using artificial neural networks. In *IEEE International Conference on Systems, Man, and Cybernetics - 2021*, pages 1254–1259, 2021.

[27] E. Nagy, M. Puskás, and D. A. Drexler. Comparison of artificial neural network and anfis for parameter estimation of a tumor model. In *IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics SAMI (2022)*, pages 133–139, 2022.

[28] M. Puskás, B. Gergics, A. Ládi, and D. A. Drexler. Parameter estimation from realistic experiment scenario using artificial neural networks. In *2022 IEEE 16th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pages 161–168, 2022.

[29] M. Siket, G. Eigner, and L. Kovács. Sensitivity and identifiability analysis of a third-order tumor growth model. In *2020 IEEE 15th International Conference of System of Systems Engineering (SoSE)*, pages 417–422, 2020.

[30] F. Cacace, V. Cusimano, and P. Palumbo. Optimal impulsive control with application to antiangiogenic tumor therapy. *IEEE Transactions on Control Systems Technology*, 28(1):106–117, Jan 2020.

[31] B. Péceli, D. A. Drexler, and L. Kovács. Optimal scheduling of low-dose metronomic chemotherapy: an in-silico analysis. In *2020 IEEE 15th International Conference of System of Systems Engineering (SoSE)*, pages 411–416, 2020.

[32] D. A. Drexler and L. Kovács. Optimization of low dose metronomic therapy based on pharmacological parameters. *IFAC-PapersOnLine*, 2021. 11th IFAC Symposium on Biological and Medical Systems.

[33] M. F. Dömény, M. Puskás, L. Kovács, and D. A. Drexler. In silico chemotherapy optimization with genetic algorithm. In *2023 IEEE 17th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pages 97–102. IEEE, May 2023.

[34] T. D. Szűcs, M. Puskás, and D. A. Drexler. Model predictive fuzzy control in chemotherapy optimization. In *IEEE 17th International Symposium on Applied Computational Intelligence and Informatics SACI 2023*, pages 103–108, 2023.

[35] M. Gluzman, J. G. Scott, and A. Vladimirsky. Optimizing adaptive cancer therapy: dynamic programming and evolutionary game theory. *Proceedings of the Royal Society B: Biological Sciences*, 287(1925):20192454, April 2020.

[36] G. A. Ezzell and L. Gaspar. Application of a genetic algorithm to optimizing radiation therapy treatment plans for pancreatic carcinoma. *Medical Dosimetry*, 25(2):93–97, 2000.

[37] Y. Yu, M. Schell, and J. Y. Zhang. Decision theoretic steering and genetic algorithm optimization: application to stereotactic radiosurgery treatment planning. *Medical Physics*, 24(11):1742–1750, 1997.

[38] S. Duraipandian, W. Zheng, J. Ng, J. J. Low, A. Ilancheran, and Z. Huang. In vivo diagnosis of cervical precancer using raman spectroscopy and genetic algorithm techniques. *Analyst*, 136(20):4328–4336, 2011.

[39] H. Bozcuk1ACEDF, U. Bilge2CDE, E. Koyuncu3AB, and H. Gulkesen2CD. An application of a genetic algorithm in conjunction with other data mining methods for estimating outcome after hospitalization in cancer patients. *Med Sci Monit*, 10(6):251, 2004.

[40] M. F. Dömény, M. Puskás, L. Kovács, and D. A. Drexler. Population-based chemotherapy optimization using genetic algorithm. In *2023 IEEE 21st International Symposium on Intelligent Systems and Informatics (SISY)*, pages 23–28. IEEE, September 2023.

[41] D. A. Drexler, T. Ferenci, A. Lovrics, and L. Kovács. Tumor Dynamics Modeling based on Formal Reaction Kinetics. *Acta Polytechnica Hungarica*, 16:31–44, 2019.

[42] J. Sápi, L. Kovács, D. A. Drexler, P. Kocsis, D. Gajári, and Z. Sápi. Tumor volume estimation and quasi-continuous administration for most effective bevacizumab therapy. *PLOS ONE*, 10:1–20, 11 2015.

[43] D. A. Drexler, J. Sápi, and L. Kovács. Modeling of tumor growth incorporating the effects of necrosis and the effect of bevacizumab. *Complexity*, pages 1–11, 2017.

[44] M. Puskás, B. Gergics, B. Gombos, A. Füredi, G. Szakács, L. Kovács, and D. A. Drexler. Noise modeling of tumor size measurements from animal experiments for virtual patient generation. In *2023 IEEE 27th International Conference on Intelligent Engineering Systems (INES)*, pages 000053–000060. IEEE, 2023.

[45] N. Sagias and G. Karagiannidis. Gaussian class multivariate weibull distributions: theory and applications in fading channels. *IEEE Transactions on Information Theory*, 51(10):3608–3619, 2005.

[46] *Anderson–Darling Test*, pages 12–14. Springer New York, New York, NY, 2008.

[47] G. Marsaglia and J. Marsaglia. Evaluating the anderson-darling distribution. *Journal of Statistical Software*, 9(2):1–5, 2004.

[48] K. Bury. *Statistical Distributions in Engineering*. Cambridge University Press, 1999.

[49] Y. Yu, X. Si, C. Hu, and J. Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270, 2019.

[50] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[51] F. Chollet et al. Keras. https://keras.io, 2015.

[52] K. Deep, K. Singh, M. Kansal, and C. Mohan. A real coded genetic algorithm for solving integer and mixed integer optimization problems. *Applied Mathematics and Computation*, 212:505–518, 06 2009.

[53] K. Deep and M. Thakur. A new mutation operator for real coded genetic algrithms. *Applied Mathematics and Computation*, 193:211–230, 10 2007.

[54] S. Chan and P. Treleaven. Chapter 5 - continuous model selection for large-scale recommender systems. In V. Govindaraju, V. V. Raghavan, and C. Rao, editors, *Big Data Analytics*, volume 33 of *Handbook of Statistics*, pages 107–124. Elsevier, 2015.

[55] O. Roeva, S. Fidanova, and M. Paprzycki. Influence of the population size on the genetic algorithm performance in case of cultivation process modelling. *2013 Federated Conference on Computer Science and Information Systems*, pages 371–376, 2013.

[56] M. Odetayo. Optimal population size for genetic algorithms: an investigation. In *IEE Colloquium on Genetic Algorithms for Control Systems Engineering*, pages 2/1–2/4, 1993.