

# Machine Learning Optimizations of BioMeld-Voxelyze Simulator

Jela Babić<sup>1,\*</sup>, Dušica Knežević<sup>1</sup>, Miloš Savić<sup>1</sup>, Vladimir Kurbalija<sup>1</sup> and Igor Balaz<sup>2</sup>

<sup>1</sup>Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Serbia

<sup>2</sup>Laboratory for Meteorology, Physics and Biophysics, Faculty of Agriculture, University of Novi Sad, Serbia

## Abstract

This paper presents the development process for a simulation software with an application in designing bio-hybrid machines (BHMs). The software's main objective is to determine optimal values of the physical properties of materials used to manufacture BHMs by utilizing a genetic search algorithm. Consequently, a huge number of potential material configurations have been analyzed in an efficient way to provide practically employable results. The primary analysis of the simulation software pointed out that the initially developed simulator has efficiency issues within the physics engine-oriented component. To improve the efficiency of the software, a regression-based machine learning approach is used as a surrogate model to predict outcomes of voxel-based simulations in the search process.

## Keywords

Bio-hybrid Machines, Voxelyze Engine, Software Improvement, Genetic Algorithm, Machine Learning

## 1. Introduction

Bio-hybrid machines (BHMs) combine living cell actuators with artificial materials in order to achieve greater autonomy, flexibility and energy efficiency compared to standard robots [1]. Although BHMs have great promise for future applications in many fields, BHM technologies are still in the early stages of development with no significant efforts made towards efficient BHM manufacturing leveraging AI-powered modeling and simulation of BHMs. This paper describes the evolution process and improvements that have been achieved during the modeling, development and optimization of a BHM simulation software.

A prerequisite for designing BHMs suitable for real-world applications is to identify and consider all potential BHM morphologies in a desired environment. To achieve this goal, we use the Voxelyze [2] modeling framework coupled with a genetic algorithm [3]. The Voxelyze physics engine simulates BHM design and deals with parameters related to the physical properties of materials from which BHMs are constructed. Relying on Voxelyze, we have developed a BioMeld-Voxelyze simulator that is part of the BioMeld modeling and simulation framework [4], whose main goal is to produce efficient designs of BHMs acting as bio-hybrid catheters [5].

Our BioMeld-Voxelyze simulator analyzes BHM's behavior in an environment in which BHM (catheter) may collide with external objects (vascular walls). It also contains a genetic search feature that selects the optimal configuration of material properties of actuating and non-actuating voxels leading to the most efficient BHM catheter displacement, while at the same time minimizing its pressure on vascular walls.

BioMeld-Voxelyze genetic search functionality may be extremely slow since the full simulation has to be performed for each chromosome encoding values of physical properties. Consequently, one may ask if performing the full simulation can be replaced by a predictive model, and, in turn, significantly

---

*SQAMIA 2024: Workshop on Software Quality, Analysis, Monitoring, Improvement, and Applications, September 9–11, 2024, Novi Sad, Serbia*

\*Corresponding author.

✉ jelab@dmi.uns.ac.rs (J. Babić); lucy@dmi.uns.ac.rs (D. Knežević); svc@dmi.uns.ac.rs (M. Savić); kurba@dmi.uns.ac.rs (V. Kurbalija); igor.balaz@df.uns.ac.rs (I. Balaz)

ORCID 0000-0001-5244-3725 (J. Babić); 0000-0002-4854-6577 (D. Knežević); 0000-0003-1267-5411 (M. Savić); 0000-0002-9599-4495 (V. Kurbalija); 0000-0002-6831-9232 (I. Balaz)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

accelerate the execution of the genetic search algorithm. Therefore, in this paper we investigate whether the genetic search functionality can be replaced by efficient machine learning surrogate models based on regression algorithms.

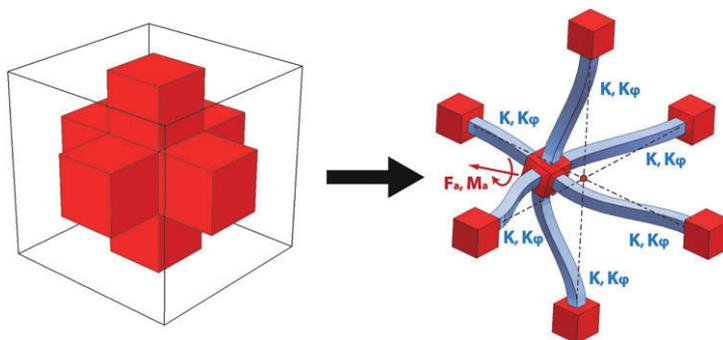
The rest of the paper is structured as follows. The BHM modeling approach is presented in Section 2. In the following Section 3 we describe the BioMeld-Voxelyze simulator. Improvements achieved by developing an alternative solution for the time-consuming module of the simulator are explained in 4. The last section concludes the paper.

## 2. BHM Modeling Approach

The first task in the modeling and simulation of BHMs is the design and analysis of their morphologies (voxel-based geometries). In our solution, the Voxelyze physics engine [2] simulates BHM designs and deals with parameters related to the physical properties of materials used to construct BHMs.

Voxelyze is a highly sophisticated physics engine designed to model and simulate the mechanical behavior of soft and flexible materials. The system uses a finite element method, where a material is divided into a mesh of interconnected elements, to model the deformation and mechanical properties of the material. The elements in the mesh represent the behavior of small volumes of the material, and their interactions with each other are used to describe the overall mechanical response of the material.

The main modeling approach in Voxelyze is based on voxels and mass-spring lattice. Each voxel in the lattice is modeled with six degrees of freedom: three translational and three rotational degrees of freedom. Furthermore, each voxel contains its mass and rotational inertia to support more complex simulations with lateral shearing and rotation. Finally, each voxel is connected to 6 neighboring voxels with connection elements, where every connection contains its own translational and rotational stiffness (Figure 1). Such a setup provides a very realistic deformation simulation under different forces and moments.



**Figure 1:** Voxel connections in Voxelyze. Visualization is taken from [2]

The task of the BioMeld-Voxelyze component in the whole pipeline is to consider and select parameters related to the physical properties of materials used to build BHM and its morphology (geometry). More precisely, in Voxelyze engine we apply force to the BHM (e.g., catheter) to move it through an environment that contains other objects (e.g., vascular walls). The Voxelyze takes care of various physical aspects including: forces, collision detection, gravity, material properties and temperature.

## 3. BHM Simulation Software

Biohybrid machines (BHMs) morphologies are created and tested in the soft-body physics engine Voxelyze on a 3D Cartesian lattice workspace, where voxels are connected together to form a functional machine. The simulator models the distance between adjacent voxels as Euler-Bernoulli beams [6]. All classes and structures originating from the base engine are effectively used. Overall, including base Voxelyze classes, the software refers to five different packages, `voxlib`, `morph`, `ga`, `camera`, and

bhmsim. To run the engine, it is required to create an instance of the CVoxelzyze class. This instance is mandatory as a base for constructing different topologies utilizing voxels. One voxel determines its own material specification (CVX MaterialVoxel) and the intervoxel material specification (CVX MaterialLink). Besides the voxel definition, it is also possible to set up other simulation variables, like gravitational acceleration and friction to voxels in contact with the surface plane, and check simulation results in several different aspects and points of simulation. As a core, Voxelyze is a library that allows different simulations at the level of voxel specifications, so the main focus of the remaining components is on customization of the core functionalities to the domain problem specification, i.e. to specify and simulate different types of objects (e.g., BHM and vascular walls) created from a set of connected voxels.

At the lowest level, close to the base Voxelyze engine, we have the bhmsim package. The idea behind this part of the BHM simulator is to provide basic structures for building different simulation scenarios in terms of morphologies, hyperparameter values and simulator goals.

Morphology classes contained in the morph package should be considered as an auxiliary tool for easier setting of the BHM simulator and simulated objects. It holds plain, low-level information about the physical properties of simulation objects. It is possible to visualize assembled morphologies through instances of classes inside the camera package. Classes defined within this package are designed to enable two sights over morphologies and their motions during simulation in two and three-dimensional space. For this purpose, The OpenGL Utility Toolkit (GLUT) [7] library was used.

Finally, the last module of our BHM simulator is intended to run a configurable genetic algorithm through numerous simulations to identify the best-featured combination of materials. The component name is ga and it contains class definitions for basic concepts of the genetic algorithm.

Generally speaking, genetic algorithms are heuristic algorithms applicable to a wide range of optimization problems with the goal of achieving a solution by simulating the mechanism of biological evolution. In preparation for building a genetic algorithm, the initial problem must be amenable to redesign in the GA framework. This framework expects to have a set of problem instances (valid problem solutions) as an initial population which is repeatedly modified until the desired condition is fulfilled. One instance of the problem has to encode properties whose values impact the optimal solution of the problem. In the case of BHM simulation, the goal is to identify properties of the materials that potentially would be suitable for the production of BHMs intended for use in real-world scenarios. In our BHM simulator, the goal is to identify the stiffness and density of the material from which BHMs are supposed to be made.

At the highest level of abstraction, the optimization problem is solved by the following steps:

- (1) generation of the initial population which produces a set of different BHMProperties assets,
- (2) creation of Voxelyze physical construction through import files with model data,
- (3) assignment of the generated properties over the physical presentation and evaluation incorporated with GA optimization process.

GA optimization is a cyclic process that involves GA operators over previously evaluated individuals. Typically used GA operators are selection, crossover, and mutation. The usage of these operators ensures the quality of the genetic evaluation process by maintaining the quality of the population in each iteration. The algorithm is performed with the evaluation of every individual genome of the population, after which the best N individuals are selected and marked for the next round of the genetic algorithm. At this point, population size is decreased and must be compensated (by creating new individuals) before the next evaluation round to match the initial population size. New individuals are created from the existing ones, through the selection of two to serve as a source of genetic material. Parents are selected randomly in the selection process over which the crossover operator is applied. Crossover creates a child individual through a combination of parents' genetic material. For this GA operator, we consider that the child genome contains an arithmetic value of the corresponding parents' attributes. The production of new individuals is finished after restocking the number of genomes in the population to the initially defined size.

Preparation of the new offspring for the next algorithm generation requires mutation over some of the individuals in the population. The mutation is performed over every individual with a certain

probability and it considers a probable slight adaptation of every observed property. The function looks for an old value from the genome of the current property and changes it by adding a randomly generated noise. As it is already pointed out, not every individual has to be mutated in the process of new offspring preparation, just has to be defined as a part of evolution. The goal of the GA implementation is to achieve a certain number of iterations and the best configuration in the latest iteration is considered as a champion (optimal solution).

In terms of the quality assessment mechanism, the fitness function is defined to maximize the BHM displacement and minimize the pressure on external objects in the environment (e.g., vascular walls). In other words, the goal is to find individuals that achieve the largest possible displacement and at the same time make the lowest pressure on the vascular wall. After every individual of the population is evaluated by computing its fitness value, the population is sorted by the obtained fitness values in order to keep the top  $k$  best configurations for the generation of new offspring.

### 3.1. Physic engine approach results

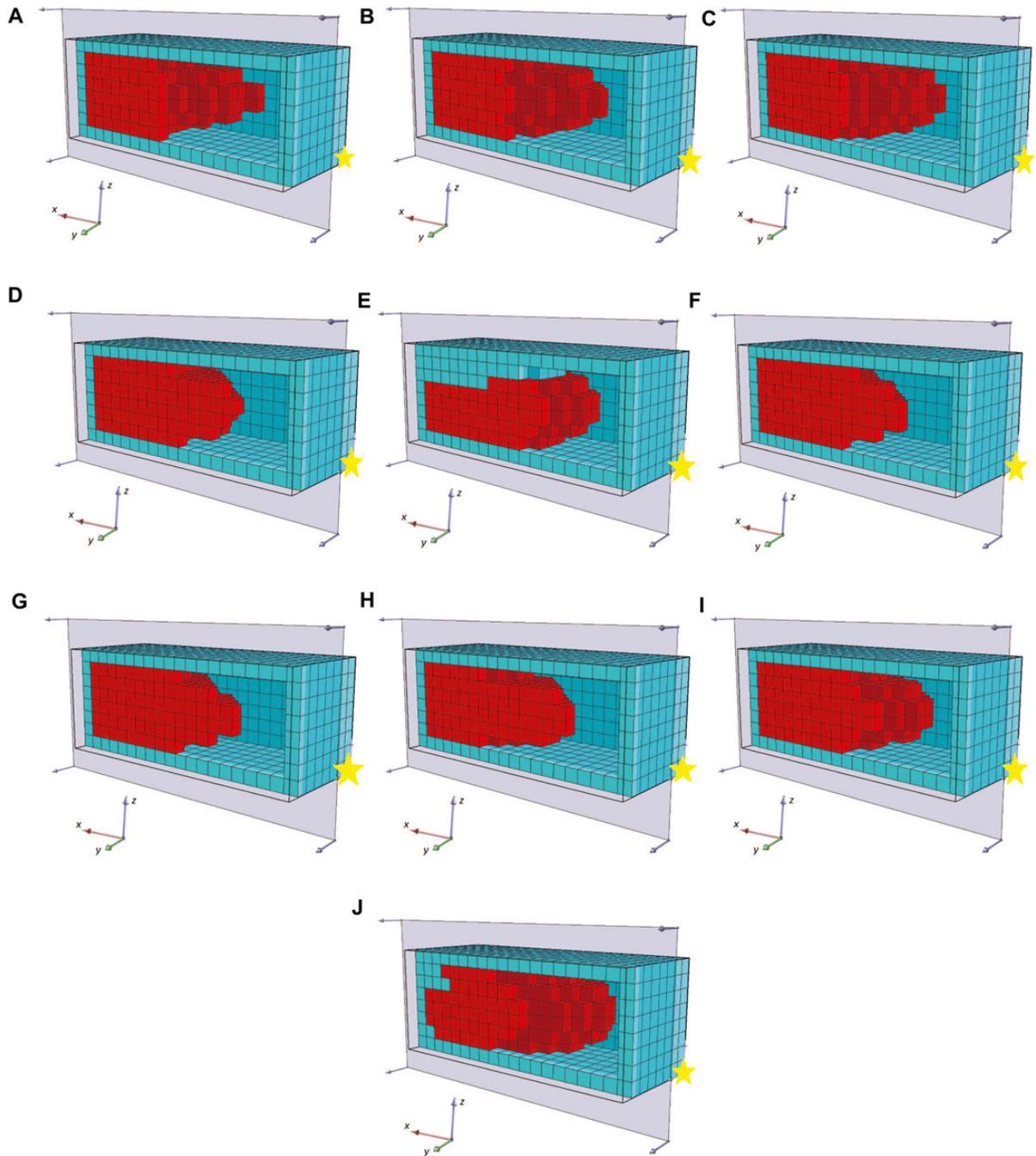
Figure 2 presents the champion morphologies produced within BioMeld modeling and simulation framework that deals with the morphology optimizations based on compositional pattern producing networks [4]. Those champion morphologies are used in the Biomeld-Voxelyze simulator for performing experiments and measuring improvements. Blue voxels represent a passive type of voxels (non-actuating), and red voxels represent active voxels (actuating voxels, muscle tissue). The simulation environment also includes one non-BHM object which is a 3D plane whose voxels have a density equal to 960 kg/m<sup>3</sup> and stiffness equal to 10MPa. A constant force is applied to the first actuating voxel of BHM towards the non-BHM object so that those two bodies collide.

For every considered morphology, GA simulation examines the space for the selection of a winning configuration. The examined space refers to the possible range of values for monitored properties of interest crucial for understanding the goal features. One considered configuration consists of stiffness and density material properties whose correlative behavior is studied through Voxelyze simulation. GA component here has the role of origination of the configurations whose analysis should be considered within the physics engine simulator. The idea was to try to have as many as possible provided combinations of values for the BHMs component building features to consider so that the defined goal is accomplished with greater certainty.

GA parameters, such as the size of the population, number of iterations, and threshold value for new generation genesis were selected by using a probabilistic method. Considering the number of possible values for the properties of BHM components, and the possible combinations, the idea was to try to achieve the highest percentage of examinations of parameter space but also to identify simulation efficiency for different sizes of problems selected to solve.

Table 1 provides perspective over executed experiments. Every row in the table corresponds to the specific morphology from Figure 2 and a corresponding simulation executed. One GA simulation is determined with GA parameters used within, the number of iterations  $N$ , population size  $P$ ,  $K$  value (GA threshold – how many instances are kept for the next GA iteration), and the longest and shortest iteration execution time. Iteration time duration is selected within the first 50 iterations of every simulation.

As it can be observed executed simulations had a very long execution time, and most of them were interrupted before the end. The analysis of the completed GA iterations identified that none of the considered experiments did not evolved its fitness function to the point of saturation. We will focus on the results of the experiment performed for configuration A. The scope for the simulation was a relatively small dimension, counting the 200 iterations of GA for the population size 80, where almost one-quarter of the population was retained for the next iteration. The time required for the longest iteration to be performed was more than six hours, whereas the shortest iteration needed only two hours less than that. Considering that, we can set 5 hours as the average time execution of one iteration, which makes this simulation have a total duration of approximately 42 hours. The most time-consuming component of the software was the calculation that a soft body physic machine Voxelyze had.



**Figure 2:** Visualization of BHM morphologies used in the evaluation of regression models predicting outcomes of simulations. The representation is taken from [4].

## 4. Machine Learning Optimizations of BioMeld-Voxelyze Simulator

To answer the above-stated research question we have examined several regression models trained to predict the outcomes of simulations participating in the cost function of the genetic search algorithms: displacement of BHM and pressure on non-BHM objects. The examined regression models are selected to have fast training and inference algorithms [8, 9]. The following regression models, whose implementation is available in the Scikit-learn Python library for machine learning [10], are evaluated for predicting the target variables:

- Linear regression (LINEAR),
- Lasso regression (LASSO),

**Table 1**

GA experiments time usage for first 50 iterations.  $N$  – number of iterations in GA,  $P$  – population size

Configuration	$N$	$P$	$K$	Longest iteration [hr]	Shortest iteration [hr]
A	200	80	30	06:19:33	04:06:31
B	300	100	40	08:20:08	04:34:54
C	1000	100	25	08:11:48	03:50:11
D	500	50	25	08:01:50	06:18:33
E	600	300	100	04:19:29	02:20:06
F	600	300	100	23:50:15	15:25:07
G	200	300	25	23:52:22	20:09:23
H	100	500	200	42:56:16	25:49:49
I	800	100	60	08:51:40	04:14:23
J	300	300	50	24:43:23	18:41:59

- Elastic net regression (ELNET),
- Support vector machine regression (SVM),
- Random forest regression (RF),
- K-nearest neighbors regression (KNN), and
- Adaboost regression (ADAB).

The linear regression algorithm determines the coefficients of a linear model by minimizing the residual sum of squares between real values of the outcome variable and predictions obtained by the linear approximation. Lasso and ElasticNet find a linear model by minimizing the residual sum of squares with incorporated regularisation penalties: Lasso uses the L1 regularisation penalty, while ElasticNet uses both L1 and L2 regularisation penalties. Regression models trained over BHM Simulator data had default values for L1 and L2 regularisation penalties, defined within Scikit-learn implementation.

SVM, RF, and KNN are adaptations of the corresponding classification algorithms for regression tasks. Support vector machine classifiers are based on the idea of using linear models to identify non-linear boundaries of categories. This is achieved by transforming data instances into a new higher-dimensional space using non-linear mapping. Quadratic programming algorithms are then employed in the higher-dimensional space to determine maximum margin hyperplanes separating instances from different categories. KNN is an instance-based (lazy) classifier: the category for a given data instance is determined from categories of the  $K$  closest instances from the training dataset determined using some distance function (e.g., Euclidean or Manhattan distance). In our experiments, we have set  $K$  to 20. A random forest is an ensemble of decision trees learned from bootstrapped samples of the training data. The random forest algorithm employs so-called feature bagging to determine a random subset of features for learning individual decision trees. The category for a given input instance is then the most frequent category derived from decision trees in the ensemble. In our experiments, we have used random forest ensembles containing 50 trees.

AdaBoost is a meta-learning algorithm boosting an arbitrary regression method (decision-tree-based regression in our case) towards data instances having high prediction errors. The algorithm starts by assigning equal weights to all instances in the training dataset. Those weights indicate how hard it is to predict the outcome variable for a particular instance, i.e., instances with higher prediction errors have higher weights. Some base regression method is employed to form an initial regression model. Instances in the training datasets are reweighted according to errors obtained by the trained regression model. Then, a new regression model is learned using a loss function that is pondered with weights. The two steps are repeated either for a fixed number of iterations or until the error of the model becomes acceptable. Predictions by AdaBoost are made considering all sequentially trained regression models.

As the baseline for evaluating the above-mentioned regression models, we use the so-called DUMMY regression model. The DUMMY model always predicts the same value: the mean of the outcome variable computed from the training dataset.

For the evaluation of models' performance, we used 10-fold cross-validation. The following metrics are used to assess the quality of predictive models:

- mean absolute error (MAE)

$$\frac{\sum_{i=1}^n |x_i - y_i|}{n},$$

- mean squared error (MSE)

$$\frac{\sum_{i=1}^n (x_i - y_i)^2}{n},$$

- coefficient of determination (R2)

$$1 - \frac{\sum_{i=1}^n (x_i - \hat{x}_i)^2}{\sum_{i=1}^n (x_i - \bar{x})^2},$$

- Pearson's correlation coefficient (PC):

$$\frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}.$$

All the measures are formally defined above so that  $n$  is the number of dataset instances,  $x_i$  is the target attribute value of  $i$ -th instance,  $y_i$  is the predicted value of the target attribute of the  $i$ -th instance,  $\bar{x}$  is the mean of the target attribute's values, and  $\bar{y}$  is the mean value of all predicted values for the target attribute. Lower MAE and MSE values indicate better-performing models (for perfect regressors MAE and MSE are equal to 0), whereas higher values of R2 and PC indicate accurate predictors (the maximal value of R2 and PC indicating perfect predictors is equal to 1).

#### 4.1. ML approach results

The evaluation of regression models used to predict the outcomes (pressure and displacement) of simulations is conducted on 10 champion morphologies [4] that are shown in Figure 2. To obtain datasets for 10-fold cross-validation, for each morphology configuration (denoted from A to J) we have performed 5000 different simulations for randomly selected input parameters in the following ranges:

- stiffness of actuating voxels is in the range [1, 20] MPa,
- density of actuating voxels is in the range [1000, 1500] kg/m<sup>3</sup>,
- stiffness of non-actuating voxels is in the range [30, 50] MPa , and
- density of non-actuating voxels is in the range [1500, 3000] kg/m<sup>3</sup>.

The result of 10-fold cross-validation for examined morphology configurations A, B, and C are presented in Table 2- 4. Tables show the estimate of mean absolute error (MAE), mean squared error (MSE), coefficient of determination (R2), and Pearson's correlation coefficient (PC) for the target variables (PRESSURE and DISPLACEMENT). It is important to mention that the PC for DUMMY cannot be computed since DUMMY models always predict the same value. Resulted values are averaged using values calculated after each iteration of the 10-fold cross-validation performed.

A regression model  $R$  can be considered as an accurate predictor for a target variable if the following conditions are satisfied:

- $MAE(R) \ll MAE(DUMMY)$ ,
- $MSE(R) \ll MSE(DUMMY)$ ,
- $R2(R) > 0.9$ , and
- $PC(R) > 0.9$ .

The symbol  $\ll$  denotes "much smaller". According to the above-stated criteria, the list of accurate predictors for PRESSURE and DISPLACEMENT is given in Table 5. As it can be observed:

**Table 2**

Evaluation of regression models for data generated from 5000 different BioMeld-Voxelyze simulations over morphology configuration A.

Conf A	PRESSURE				DISPLACEMENT			
	MAE	MSE	R2	PC	MAE	MSE	R2	PC
DUMMY	2732.45	11684957.52	-0.0020	nan	1.5116	3.3441	-0.0012	nan
LINEAR	1895.58	6597637.57	0.4334	0.6628	0.4500	0.4311	0.8705	0.9336
LASSO	1996.08	7225193.92	0.3781	0.6289	0.5600	0.8787	0.7352	0.8655
ELNET	1996.09	7225191.55	0.3781	0.6289	0.5600	0.8787	0.7352	0.8655
SVM	2558.56	11234923.51	0.0364	0.4775	0.1111	0.0332	0.9901	0.9954
RF	136.46	103318.42	0.9911	0.9956	0.0163	0.0017	0.9995	0.9998
KNN	483.08	789951.89	0.9322	0.9657	0.0394	0.0054	0.9984	0.9992
ADAB	1136.86	1933317.83	0.8343	0.9280	0.0942	0.0158	0.9953	0.9979

**Table 3**

Evaluation of regression models for data generated from 5000 different BioMeld-Voxelyze simulations over morphology configuration B.

Conf B	PRESSURE				DISPLACEMENT			
	MAE	MSE	R2	PC	MAE	MSE	R2	PC
DUMMY	1119.01	2281736.01	-0.0014	nan	0.2386	0.0789	-0.0026	nan
LINEAR	944.73	1842950.52	0.1915	0.4402	0.0544	0.0045	0.9428	0.9712
LASSO	1032.69	2105625.45	0.0784	0.2846	0.0923	0.0128	0.8368	0.9161
ELNET	1032.69	2105625.45	0.0784	0.2846	0.0923	0.0128	0.8368	0.9161
SVM	954.75	2029546.43	0.1120	0.4608	0.0402	0.0024	0.9692	0.9884
RF	105.23	72200.94	0.9672	0.9837	0.0033	0.0000	0.9994	0.9997
KNN	520.41	846764.25	0.6280	0.7943	0.0136	0.0005	0.9939	0.9970
ADAB	671.83	733393.43	0.6765	0.8987	0.0199	0.0006	0.9921	0.9972

**Table 4**

Evaluation of regression models for data generated from 5000 different BioMeld-Voxelyze simulations over morphology configuration C.

Conf C	PRESSURE				DISPLACEMENT			
	MAE	MSE	R2	PC	MAE	MSE	R2	PC
DUMMY	2552.05	11276240.49	-0.0012	nan	3.2549	16.0399	-0.0024	nan
LINEAR	2203.98	7738828.21	0.3121	0.5599	1.5248	3.9920	0.7505	0.8670
LASSO	2271.87	7902469.02	0.2972	0.5469	1.4800	4.3643	0.7281	0.8539
ELNET	2271.87	7902468.64	0.2972	0.5469	1.4800	4.3643	0.7281	0.8539
SVM	2416.03	10572976.66	0.0613	0.6028	0.6185	1.3116	0.9183	0.9683
RF	165.49	132120.31	0.9882	0.9941	0.0381	0.0089	0.9994	0.9997
KNN	559.31	942613.25	0.9163	0.9573	0.0792	0.0240	0.9985	0.9993
ADAB	1321.70	2568136.02	0.7716	0.8948	0.2366	0.0958	0.9940	0.9974

- RF is always an accurate predictive model for both PRESSURE and DISPLACEMENT.
- Besides RF, PRESSURE can be accurately predicted by KNN for 8 morphology configurations and by ADAB for 4 configurations.
- DISPLACEMENT can be accurately predicted by RF, KNN, and ADAB in all examined configurations.
- LINEAR and SVM models have a minor accurate power (for some configurations they can

accurately predict DISPLACEMENT), while ELNET and LASSO are always bad predictors of considered target variables.

**Table 5**

Summary of the success of providing an accurate prediction per feature of interest (PRESSURE and DISPLACEMENT) for the examined models (DUMMY, LINEAR, LASSO, ELNET, SVM, RF, KNN, ADAB) in the considered configurations (A-J).

Configuration	Accurate predictors for PRESSURE	Accurate predictors for DISPLACEMENT
A	RF, KNN	SVM, RF, KNN, ADAB
B	RF	LINEAR, SVM, RF, KNN, ADAB
C	RF, KNN	SVM, RF, KNN, ADAB
D	RF, KNN, ADAB	RF, KNN, ADAB
E	RF, KNN, ADAB	RF, KNN, ADAB
F	RF, KNN, ADAB	SVM, RF, KNN, ADAB
G	RF, KNN, ADAB	RF, KNN, ADAB
H	RF	LINEAR, SVM, RF, KNN, ADAB
I	RF, KNN	RF, KNN, ADAB
J	RF, KNN	SVM, RF, KNN, ADAB

## 5. Conclusion

The main focus of this paper is on the design of the BioMeld-Voxelyze Simulator software, which is part of a framework for designing bio-hybrid machines (BHMs). The main purpose of the simulator is to determine optimal values of parameters related to the physical properties of materials (stiffness and density of actuating and non-actuating voxels) used to build BHM. To achieve this goal, a huge space of potential BHM configurations has to be examined.

Initially, the combination of a genetic algorithm and simulations based on the Voxelyze engine was used to identify optimal BHM configurations. However, performing BHM simulations required a lot of time to process and simulate the mechanical behavior of the materials assigned to BHM components within the simulation.

To overcome the time usage problem and accelerate the execution of the genetic search algorithm, we have devised a different approach. By employing the results of the initially proposed solutions, regression models were built to predict the outcomes (pressure and displacement) of simulations conducted on 10 champion morphologies. For the evaluation of models' performance, we used 10-fold cross-validation and a set of metrics to assess the quality of predictive models.

Our experimental results indicate that random-forest regression models can accurately predict the outcomes of simulations. This result has an important practical consequence: the genetic search algorithm of the BioMeld Voxelyze simulator looking for optimal physical properties of BHM voxels can be significantly accelerated by using regression predictive models instead of performing simulations for each candidate chromosome encoding physical properties.

## Acknowledgments

This project has received funding from the European Union's Horizon Europe research and innovation program under grant agreement No.101070328.

## References

- [1] L. Ricotti, B. Trimmer, A. W. Feinberg, R. Raman, K. K. Parker, R. Bashir, M. Sitti, S. Martel, P. Dario, A. Menciassi, Biohybrid actuators for robotics: A review of devices actuated by living cells, *Science Robotics* 2 (2017) eaaq0495. doi:10.1126/scirobotics.aaq0495.
- [2] J. Hiller, H. Lipson, Dynamic simulation of soft multimaterial 3d-printed objects, *Soft Robotics* 1 (2014) 88–101. doi:10.1089/soro.2013.0010.
- [3] O. Kramer, *Genetic Algorithms*, Springer International Publishing, Cham, 2017, pp. 11–19. doi:10.1007/978-3-319-52156-5\_2.
- [4] M.-A. Tsompanas, I. Balaz, Outline of an evolutionary morphology generator towards the modular design of a biohybrid catheter, *Frontiers in Robotics and AI* 11 (2024). doi:10.3389/frobt.2024.1337722.
- [5] C. Salvatori, D. Trucco, I. Niosi, L. Ricotti, L. Vannozzi, A novel steerable catheter controlled with a biohybrid actuator: A feasibility study, in: *Biomimetic and Biohybrid Systems*, Springer Nature Switzerland, Cham, 2023, pp. 378–393.
- [6] O. A. Bauchau, J. I. Craig, *Euler-Bernoulli beam theory*, Springer Netherlands, Dordrecht, 2009, pp. 173–221. doi:10.1007/978-90-481-2516-6\_5.
- [7] M. Woo, J. Neider, T. Davis, D. Shreiner, *OpenGL programming guide: the official guide to learning OpenGL, version 1.2*, Addison-Wesley Longman Publishing Co., Inc., 1999.
- [8] M. Savic, V. Kurbalija, M. Ilic, M. Ivanovic, D. Jakovetic, A. Valachis, S. Autexier, J. Rust, T. Kosmidis, The application of machine learning techniques in prediction of quality of life features for cancer patients, *Computer Science and Information Systems* 20 (2023) 381–404. doi:10.2298/CSIS220227061S.
- [9] M. Savic, V. Kurbalija, M. Ilic, M. Ivanovic, D. Jakovetic, A. Valachis, S. Autexier, J. Rust, T. Kosmidis, Analysis of machine learning models predicting quality of life for cancer patients, in: *MEDES '21: Proceedings of the 13th International Conference on Management of Digital EcoSystems*, Virtual Event, Tunisia, November 1 - 3, 2021, ACM, 2021, pp. 35–42. doi:10.1145/3444757.3485103.
- [10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* 12 (2011) 2825–2830.