Synthetic Data Generation and Comparative Evaluation of Machine Learning Models for Predicting the Biomechanical Behavior of the Human Cornea

LUIZ CARLOS BRANDÃO JUNIOR[1]
RICARDO RODRIGUES MAGALHÃES[1]

UFLA - Universidade Federal de Lavras
DCC - Departmento de Ciência da Computação
P.O. Box 3037 - Campus da UFLA 37200-000 - Lavras (MG)- Brazil
[1] (luiz.junior11@estudante.ufla.br)
[2] (ricardorm@ufla.br)

Abstract. The analysis of the biomechanical behavior of the human cornea is crucial for various clinical applications, such as planning refractive surgeries and diagnosing diseases like keratoconus. Numerical simulations, like the Finite Element Method (FEM), are powerful tools for these analyses but can be computationally expensive, limiting extensive parameter explorations or real-time optimizations. Surrogate models based on Machine Learning (ML) emerge as a promising alternative but require representative datasets for training. This study details a methodology for generating synthetic data (N=10,000) to replicate the functional relationships and variability observed in a previous FEM simulation study of corneal mechanical behavior under different mesh configurations and loading conditions [1]. The performance of 19 ML regression algorithms was comparatively evaluated in predicting three key variables: the Simulated Corneal Displacement, the simulation Processing Time, and the Simulation Percentage Error (SSE). Predictor variables included applied pressure, (simulated) patient age, and FEM mesh characteristics (number of nodes and elements). The results demonstrated that models based on tree ensembles (Extra Trees, Random Forest, LightGBM, XGBoost) and KNeighbors Regressor achieved exceptional performance in predicting the simulated displacement ($R^2 > 0.999, RMSE < 0.001mm$), validating their potential as accurate surrogate models. For the time and error variables, inherently linked to the discrete scenario configuration in the synthetic data, several models reached $R^2 \approx 1.0$. The research validates the approach of generating simulation-informed synthetic data as an effective method to enable the robust training and selection of ML models for complex problems in computational biomechanics.

Keywords: Machine Learning, Biomechanics, Human Cornea, Synthetic Data, Finite Element Method, Surrogate Model, Performance Prediction, Regression, Numerical Simulation.

(Received July 7th, 2025 / Accepted December 19th, 2025)

1 Introduction

The human cornea, a transparent and avascular structure located at the anterior portion of the eye, plays a pivotal role in both the optical system and the mechanical integrity of the eyeball [3]. Its curvature and elasticity directly affect refractive power, while its biomechanical properties govern how it deforms under intraocular pressure and external forces. Alterations in these properties are closely linked to pathological conditions, such as keratoconus [4], and are crucial for the planning, safety, and long-term success of refractive surgical procedures like LASIK and PRK. Furthermore, corneal biomechanics underpin the development

and calibration of diagnostic technologies, including non-contact tonometry and elastography-based imaging methods [5, 6].

1.1 Background and Motivation

Understanding the mechanical behavior of the human cornea under various physiological and experimental conditions is essential for advancing diagnostic tools, treatment planning, and personalized medicine in ophthalmology. Among the available computational approaches, the Finite Element Method (FEM) has become the standard for simulating corneal biomechanics [7]. FEM provides detailed insights into the distribution of stress, strain, and displacement when the cornea is subjected to intraocular pressure or external loads (e.g., air-puff tonometry), while incorporating anatomically accurate geometries, realistic boundary conditions, and complex material properties such as nonlinearity, anisotropy, and viscoelasticity [8].

Despite its predictive power, FEM suffers from high computational demands. Accurate simulations, especially those employing three-dimensional models with high-resolution meshes, may require significant processing time and resources [1, 9]. This computational cost becomes a bottleneck in scenarios that require a large number of simulations, such as parametric sensitivity analyses, design optimization, virtual patient studies, or real-time decision-making in clinical settings.

To address these limitations, surrogate modeling strategies based on Machine Learning (ML) have gained prominence in recent years [10]. ML regression models, once trained, are capable of approximating the complex input–output mappings learned from FEM simulations with a fraction of the computational effort. These surrogates can be integrated into interactive platforms, automated design workflows, or real-time systems, offering nearly instantaneous predictions.

However, the effectiveness of ML models critically depends on the availability of high-quality, diverse, and sufficiently large training datasets. Relying solely on FEM to generate such datasets may negate the computational advantages sought through surrogate modeling. Each additional data point can require minutes to hours of simulation time, particularly for detailed biomechanical problems.

A promising approach to mitigate this challenge involves the use of *synthetic data*—artificially constructed datasets that replicate the statistical characteristics and physical relationships observed in simulated or experimental data [11]. By analyzing the behavior of a limited set of representative FEM simulations, it is possible to model functional dependencies and generate large volumes of data computationally. When carefully designed, these synthetic datasets preserve the essential biomechanics of the original system and serve as effective training grounds for machine learning models, enabling efficient exploration of surrogate modeling without incurring excessive computational costs.

A foundational study by Almeida and Magalhães [1] explored the influence of FEM mesh resolution on the accuracy and performance of corneal biomechanical simulations. By varying the number of nodes and elements, the authors investigated how mesh complexity affects the displacement predicted at the corneal apex under pressure, along with simulation time and error relative to experimental data. Their dataset offers a valuable basis for modeling corneal mechanical behavior and evaluating simulation trade-offs.

Building upon these insights, the present work proposes a methodology that leverages synthetic data generation to support the development of ML-based surrogate models for FEM simulations of the cornea. Specifically, this study aims to: (i) Generate a synthetic dataset ($N = 10{,}000$) that reproduces the functional relationships reported by [1], augmented with age as a simulated patient-specific parameter; (ii) Train and compare the predictive performance of 19 regression algorithms in estimating three key simulation outputs: corneal apex displacement, processing time, and simulation percentage error (SSE); (iii) Analyze the results using standard regression metrics (R^2, MAE, RMSE) and visual tools; and (iv) Discuss the feasibility of using ML models trained on synthetic data as reliable and efficient substitutes for traditional FEM simulations in computational biomechanics.

By demonstrating that surrogate models can accurately reproduce FEM outcomes with minimal computational effort, this research contributes to the advancement of real-time, data-driven approaches in ophthalmic biomechanics and highlights the potential of synthetic data in accelerating model development and testing.

1.2 Related Work

The application of numerical simulation techniques, particularly the Finite Element Method (FEM), has become central in the study of corneal biomechanics. Over the past two decades, researchers have employed FEM to simulate corneal deformation under various conditions, incorporating geometrical fidelity, nonlinear material properties, and realistic boundary constraints [7, 8]. These models have enabled important insights into the mechanical behavior of the cornea

during intraocular pressure variation, surgical interventions, and diagnostic tests such as air-puff tonometry. However, the computational cost of FEM simulations remains a barrier to broader clinical application, especially in scenarios requiring real-time or high-throughput analysis.

In parallel, the growth of data-driven techniques has inspired new approaches to approximate physical simulations using Machine Learning (ML). In biomedical engineering and biomechanics, ML has been explored to emulate simulation outputs, optimize design parameters, and predict patient-specific outcomes [10]. Specifically in ophthalmology, recent studies have applied ML to estimate corneal stiffness [6], detect keratoconus [4], and support surgical planning. Despite these advances, the integration of ML with FEM in corneal modeling remains relatively underexplored, with few works addressing how ML models can serve as efficient surrogates for computationally expensive biomechanical simulations.

A key challenge in training machine learning (ML) models for this application is the scarcity of high-quality training data. Finite element method (FEM) simulations present two major limitations:

- High computational cost
- Significant time requirements

These constraints restrict the generation of large datasets necessary for robust model development.

To overcome these challenges, synthetic data generation strategies have emerged in related domains:

- Fluid dynamics
- Structural mechanics
- Material science

These approaches [11]:

- Preserve the essential physics of the problem
- Enable creation of large training corpora
- Reduce computational costs significantly

Current research demonstrates that such methods can:

- Accelerate training of ML surrogate models
- Enhance generalization capabilities

In corneal biomechanics research, the study by Almeida and Magalhães [1] established fundamental advances in computational corneal modeling using the Finite Element Method (FEM). Their systematic investigation examined:

- The impact of FEM mesh resolution (quantified by node and element counts)
- The relationship between simulation accuracy and computational time

Through comparative analysis of:

- Predicted corneal apex displacement under internal pressure
- Experimental reference values

the authors precisely quantified the fidelity-cost trade-offs inherent in FEM simulations. This parametric analysis framework provides essential insights for:

- Clinical applications of corneal modeling
- Research methodology development

Although the original dataset produced by Almeida and Magalhães was relatively limited in size and scope, it presents a well-defined and interpretable framework that captures key biomechanical principles, making it a valuable starting point for more advanced data-driven modeling strategies. In particular, it lends itself well to the generation of synthetic datasets that preserve the functional relationships and trends observed in the original simulations. Such synthetic expansion can dramatically increase the volume of training data available for machine learning (ML) algorithms without requiring the costly execution of thousands of new FEM runs.

Despite the growing interest in combining FEM and ML for biomechanical modeling, to our knowledge, no prior work has proposed or validated a framework that (i) generates large-scale synthetic datasets informed by FEM results in corneal biomechanics, and (ii) systematically benchmarks a broad range of ML regression models in this context. Moreover, while ML applications in ophthalmology have explored disease classification, parameter estimation, and diagnostic prediction, the use of ML to emulate full-field biomechanical simulation outputs remains underexplored.

Addressing this gap, the present study proposes a novel and comprehensive methodology that bridges FEM simulations, synthetic data generation, and ML-based surrogate modeling in the specific domain of corneal biomechanics. The main contributions of this paper are:

1. **Synthetic Data Generation:** We introduce a method for generating a large-scale synthetic dataset ($N = 10{,}000$) derived from the relationships observed in the FEM simulations of

Almeida and Magalhães [1]. The dataset integrates key biomechanical input parameters, including intraocular pressure, patient age (simulated), and mesh resolution characteristics, to reproduce outputs such as corneal apex displacement, simulation time, and error.

2. **Comparative Model Evaluation:** We conduct a systematic comparison of 19 state-of-the-art machine learning regression algorithms, evaluating their ability to predict the three target variables using standard performance metrics. This evaluation enables the identification of models that are both accurate and computationally efficient for use in surrogate modeling.

3. **Validation of Surrogate Models**: We demonstrate that ensemble-based models—particularly `ExtraTreesRegressor` and `LightGBM`—can achieve exceptional predictive performance ($R^2 > 0.999$) in estimating corneal displacement. This validates their utility as accurate, computationally efficient alternatives to traditional FEM simulations.

4. **Visual and Qualitative Analysis**: Beyond quantitative metrics (R^2, MAE, RMSE), we provide:

    - Visual comparisons of predicted vs. expected deformation curves
    - Assessment of prediction smoothness and stability

    These factors are particularly crucial for clinical reliability and diagnostic applications.

The remainder of this article is organized as follows. Section 2 details the methodology employed, including the synthetic data generation process, variable selection, the ML algorithms tested, and the evaluation metrics. Section 3 presents the quantitative and qualitative results obtained, comparing the models' performance on the three prediction tasks and including a detailed visual validation. Section 4 discusses the implications of the findings, analyzing the effectiveness of the approach, the specific performance characteristics of each model, and the study's limitations. Finally, Section 5 summarizes the main conclusions and points to directions for future work.

2 Methods

2.1 Reference Data Source

The foundation for the synthetic dataset construction was the study conducted by Almeida and Magalhães [1], which employed commercial finite element software (Abaqus®) to simulate the inflation test of human corneas under varying mesh configurations. The authors systematically explored how mesh resolution affects simulation accuracy and performance by creating nine distinct FEM models, termed *Scenarios 1–9*, each characterized by different numbers of concentric rings and layers. This resulted in meshes containing between 57 and 938 nodes, and between 48 and 864 elements.

The following quantitative and qualitative data were extracted from the original publication:

- The number of nodes and elements for each FEM scenario (Table 1 in [1]).
- The corresponding simulation processing times (Figure 7 and Table 2 in [1]).
- The mean squared percentage error (SSE%) comparing each simulation's displacement curve with experimental data (Figure 10 and Table 2).
- Simulated displacement curves at the corneal apex for varying pressures (0–45 mmHg), available for a subset of scenarios (Figures 8 and 9).
- The target experimental displacement curve (Figure 2).
- A complementary analysis of the influence of patient age (50 to 90 years) on corneal displacement for Scenario 6 (Figure 11).

These data served as the empirical basis for modeling functional relationships between input conditions (mesh configuration, pressure, patient age) and output responses (displacement, error, processing time), which were subsequently used to generate synthetic data at scale.

2.2 Synthetic Data Generation

To overcome the limited size of the original dataset and enable robust training of machine learning models, we developed a synthetic data generation pipeline using `Python` v3.13, leveraging `Pandas` [12] and `NumPy` [13]. A total of $N = 10{,}000$ synthetic samples were generated, each simulating a unique combination of physiological and computational conditions. The process was deterministic, relying on parametric modeling derived from the original study, with variability introduced via randomized combinations of inputs. The generation process involved the following steps:

1. **Scenario Initialization:** For each of the 9 FEM scenarios, we stored fixed values corresponding to the number of nodes, number of elements, processing time, and SSE%. These values were extracted directly from Tables 1 and 2 in [1] and served as fixed properties associated with each `Scenario_ID`.

2. **Target Curve Modeling:** The target experimental displacement curve (Figure 2) was digitized by visually estimating discrete pressure-displacement pairs, which were then fitted to a third-degree polynomial using `numpy.polyfit`. This yielded the function `get_target_displacement(pressure)`, which returns an interpolated displacement value for any applied pressure. The function was adjusted to ensure non-negative outputs and asymptotic stability at high pressures.

3. **Age Effect Modeling:** Based on Figure 11, which shows reduced displacement in older patients (interpreted as increased corneal stiffness), we defined `get_age_factor(age)` as a linear interpolation function. This yields a multiplicative factor ranging from 1.0 for patients aged ≤ 50 to approximately 0.8 for patients aged ≥ 90. This effect was assumed to generalize across all mesh scenarios.

4. **Simulated Displacement Modeling:** The function `get_simulated_displacement(pressure, scenario_id, age)` was implemented to estimate the corneal apex displacement under given conditions. The steps are:

    - Compute the target displacement using the polynomial fit.
    - Derive a deviation from the target value, modeled as proportional to both the scenario's SSE% and the magnitude of the target displacement. A pressure-dependent scaling term was also included to replicate the curvature patterns in Figures 8 and 9. The deviation sign was assumed to be predominantly negative, reflecting the underestimation behavior seen in the original simulations.
    - Apply the age scaling factor to the deviated result.
    - Enforce a non-negativity constraint to avoid physiologically implausible outputs.

    Notably, *no explicit random noise* was injected into the displacement values to maintain smoothness and reproducibility.

5. **Sample Generation:** For each of the 10,000 samples, the following steps were performed:

    - A `Scenario_ID` (1–9), `Patient_Age_Years` (uniformly sampled from 50 to 90), and `Applied_Pressure_mmHg` (uniformly sampled from 0 to 45) were randomly generated.
    - The scenario-specific parameters (`Num_Nodes`, `Num_Elements`, `Processing_Time_s`, `SSE_Error_Percentage`) were retrieved.
    - The functions above were used to calculate `Target_Displacement_mm` and `Simulated_Displacement_mm`.
    - All variables were stored as a dictionary and appended to a list.

6. **Data Structuring and Export:** The list of samples was converted into a `Pandas` DataFrame. Columns were organized, numeric values were rounded to suitable precision, and the complete dataset was exported as a spreadsheet file in '.xlsx' format for further analysis.

The final dataset contained the following structured columns:

- `Sample_ID` – Unique identifier for each synthetic sample.
- `Scenario_ID` – Integer (1–9) indicating FEM mesh configuration.
- `Num_Nodes` – Number of nodes in the FEM mesh.
- `Num_Elements` – Number of elements in the FEM mesh.
- `Patient_Age_Years` – Simulated patient age (50–90 years).
- `Applied_Pressure_mmHg` – Simulated intraocular pressure (0–45 mmHg).
- `Target_Displacement_mm` – Displacement expected from the experimental curve.

- `Simulated_Displacement_mm` – Displacement estimated by the FEM scenario and age effect.
- `SSE_Error_Percentage` – Error relative to the target curve.
- `Processing_Time_s` – Estimated simulation time in seconds.

2.3 Variable Selection for ML Modeling

To develop effective machine learning (ML) models capable of approximating the outcomes of finite element simulations, it is essential to define the prediction tasks and select relevant input features that are both informative and computationally accessible. Based on the structure of the synthetic dataset and the study objectives, three distinct regression tasks were formulated:

- **Task 1: Prediction of Simulated Corneal Displacement (`Simulated_Displacement_mm`)**
  This is the primary regression task, as it directly relates to the physiological output of interest in FEM simulations.
  - **Input Features:**
    - `Applied_Pressure_mmHg` — intraocular pressure applied during the simulation.
    - `Patient_Age_Years` — simulated patient age, which influences corneal stiffness.
    - `Num_Nodes` — number of nodes in the FEM mesh.
    - `Num_Elements` — number of elements in the FEM mesh.
- **Task 2: Prediction of Simulation Processing Time (`Processing_Time_s`)**
  This task aims to model the computational cost associated with each FEM configuration, which is a key factor in clinical and real-time applications.
  - **Input Features:**
    - `Num_Nodes`
    - `Num_Elements`
- **Task 3: Prediction of Simulation Error (`SSE_Error_Percentage`)**
  This task seeks to estimate the deviation of the simulated displacement from the experimental target, providing a surrogate measure of model fidelity.
  - **Input Features:**
    - `Num_Nodes`
    - `Num_Elements`

The column `Target_Displacement_mm`, which represents the experimentally derived displacement, was retained in the dataset exclusively for reference and visual validation but was not used as a target variable in any ML training task. The modeling efforts focused entirely on predicting the simulated outcomes generated via the synthetic data generation methodology.

2.4 Regression Models and Evaluation

A total of 19 distinct regression algorithms were selected to represent a broad spectrum of ML modeling strategies, encompassing linear, non-linear, tree-based, and neural architectures. These models included:

- **Linear models:** Linear Regression, Ridge, Lasso, ElasticNet
- **Instance-based models:** KNeighbors Regressor
- **Support Vector Machines:** SVR (with RBF kernel)
- **Tree-based models:** Decision Tree, Random Forest, Extra Trees
- **Boosting models:** AdaBoost, Gradient Boosting, XGBoost [15], LightGBM [16]
- **Neural network models:** MLP Regressor (Multi-Layer Perceptron)
- **Others:** Bayesian Ridge, Huber Regressor, Passive Aggressive, etc.

These models were implemented using the `scikit-learn` [14], `XGBoost` [15], and `LightGBM` [16] libraries. Default hyperparameters were used for most regressors, unless otherwise specified in Section 3.

For each of the three defined regression tasks, the following pipeline was systematically executed:

1. **Train/Test Split:** The dataset was partitioned into training and testing subsets using an 80/20 split. The `random_state` parameter was fixed at 42 to ensure reproducibility.
2. **Feature Scaling:** All numerical input features were standardized using the `StandardScaler` from `scikit-learn`, fitted exclusively on the training data to prevent data leakage.

3. **Model Training and Evaluation:** A loop was implemented to train each of the 19 regressors on the scaled training set and evaluate their performance on the scaled test set.

4. **Performance Metrics:** Four evaluation metrics were computed on the test set:

   - Coefficient of determination (R^2)
   - Mean Absolute Error (MAE)
   - Mean Squared Error (MSE)
   - Root Mean Squared Error (RMSE)

The model outputs were subsequently ranked and compared across tasks, and the best-performing models were selected for further qualitative and visual analysis in Section 3.

2.5 Synthetic Dataset Generation Algorithm

To complement the textual description of the synthetic data generation process, we provide in Algorithm 1 a formalized pseudocode representation of the methodology. This structured algorithm synthesizes how key information from the FEM simulation study by Almeida and Magalhães [1] was incorporated into the synthetic modeling process. The algorithm also outlines how deterministic functions were implemented to mimic realistic biomechanical behaviors, incorporating input parameters such as mesh configuration, intraocular pressure, and patient age to generate a large-scale dataset with $N = 10{,}000$ samples.

Algorithm 1 Synthetic Cornea Dataset Generation

GenerateCorneaSyntheticData Set `output_path` (Drive if available, else local) Load 9 scenarios (nodes, elements, time, SSE) Load target curve (P, D) and fit cubic polynomial $poly\_t$ Define age factor f_a:

$$f_a = \begin{cases} 1.0, & idade \leq 50 \\ 0.8, & idade \geq 90 \\ \text{linear}(1.0, 0.8), & \text{otherwise} \end{cases}$$

SimDisplacement$p, s, idade$ $d_t \leftarrow poly\_t(p)$ $f \leftarrow (\text{SSE}_s/5)/100$ $d \leftarrow (d_t - d_t f(p/45)) \cdot f_a$ **return** $\max(0, d)$ $N \leftarrow 10000$, initialize dataset **for** $i = 1$ *to* N **do**-_

Randomly select scenario, $idade \in [50, 90]$, $p \in [0, 45]$ Compute simulated displacement Append record to dataset

Build DataFrame and save as Excel

This algorithm ensures reproducibility and interpretability of the data synthesis pipeline. Each step encapsulates a clear logic grounded in domain knowledge and empirical evidence. Moreover, by avoiding the addition of random noise and relying on deterministic functions derived from real data, the generated dataset preserves coherence and enables stable model training and evaluation.

2.6 Visual Validation of Displacement Predictions

Beyond quantitative metrics such as R^2, MAE, and RMSE, visual inspection of predicted curves plays a critical role in assessing the quality and reliability of surrogate models in biomechanical contexts. In particular, for problems involving the mechanical deformation of biological tissues, the shape and smoothness of predicted displacement-pressure curves must reflect realistic physical behavior, aligning with known experimental trends.

After evaluating all models for predicting synthetic corneal displacement, we selected the top-performing regression model—typically a tree-based ensemble (*XGBoost* or *Extra Trees*)—to generate comparative visualizations. These plots enable:

- Visual assessment of how closely the model approximates expected corneal biomechanical responses across varying input conditions
- Evaluation of model generalization:
  - Across different FEM simulation scenarios
  - For various patient age groups

Two key visualizations were generated:

- **Scenario Comparison Plot:** For a fixed patient age, displacement curves were plotted across several FEM mesh configurations (scenarios with different numbers of nodes and elements). This emulates the comparison presented in Figure 9 of [1], allowing for a qualitative check on whether mesh complexity (reflected in simulation error) leads to systematically different predictions.
- **Age Effect Plot:** For a fixed scenario, displacement curves were plotted across different patient ages (e.g., from 50 to 90 years). This emulates the analysis from Figure 11 of [1], evaluating whether the trained model correctly captures the reduction in tissue deformability with increasing age.

The complete plotting procedure—comprising model retraining, pressure curve definition, input

generation, prediction, and visualization—is formally specified in Algorithm 2. All plots were automatically saved to either:

- The configured Google Drive folder, or
- A local file path (when cloud storage was unavailable)

This automated saving mechanism ensures:

- Full reproducibility of results
- Seamless integration into model validation workflows

Algorithm 2 Line Plot Generation: Predicted vs. Target Displacement

GenerateDisplacementPlots 1. Select high-performance regression model Choose regression model (e.g., XGBoost) Define main prediction task: displacement Select input features and target variable Split data into training and test sets Scale training data with `StandardScaler` 2. Train selected model Fit model on scaled training data 3. Define polynomial target displacement curve **if** *predefined polynomial function available* **then**-_

_ Use coefficients to build target curve **else**

_ Estimate target curve from data by averaging Use interpolation for target curve

4. Plot 1: Scenario Comparison (e.g., Figure 9) Define set of scenarios with varying mesh sizes Fix patient age for all curves **for** *each selected scenario* **do**

_ Create input DataFrame with pressure range and scenario parameters Scale input using previously fitted scaler Predict displacement using trained model Plot predicted curve Plot target curve Configure and save plot to Google Drive

5. Plot 2: Age Effect (e.g., Figure 11) Select fixed scenario (e.g., Scenario 5) Define list of patient ages **for** *each age in age list* **do**

_ Create input DataFrame with pressure range and fixed scenario Scale input using previously fitted scaler Predict displacement using trained model Plot predicted curve for that age Configure and save plot to Google Drive

Display completion message

2.7 Comprehensive Regression Pipeline for Biomechanical Prediction

To perform a systematic and reproducible evaluation of multiple regression algorithms in the context of biomechanical modeling, we implemented an end-to-end machine learning pipeline that integrates training, evaluation, ranking, and visualization procedures. This automated framework, described in Algorithm 3, supports multi-task regression analysis over synthetic datasets and facilitates detailed model comparisons using both statistical and graphical outputs.

The pipeline was designed to evaluate the predictive performance of approximately 20 regression models across three primary biomechanical output variables: (1) simulated corneal displacement, (2) simulation processing time, and (3) simulation error (SSE%). For each task, the appropriate input features were selected (as described in Section 2.4), and each model was trained using a standard 80/20 split and evaluated with conventional regression metrics: coefficient of determination (R^2), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

After all models were evaluated on a given task, a performance ranking table was generated, and R^2 scores were visualized using horizontal bar plots for easier interpretation and comparison. These plots were automatically saved in a dedicated results directory, either on Google Drive or in a local fallback path, depending on system configuration.

For the specific case of corneal displacement prediction—arguably the most clinically relevant task—additional steps were implemented. Models that achieved excellent accuracy ($R^2 > 0.99$) were identified as high-performing surrogates and selected for further qualitative validation. These models were retrained on the full training dataset and used to generate two types of displacement-pressure curves:

- **Scenario Comparison Plot:** For a fixed patient age, predictions were generated across multiple mesh configurations (i.e., different FEM scenarios), and compared against the reference target curve. This analysis helps assess whether the model can replicate the expected differences in behavior due to mesh resolution.
- **Age Effect Plot:** For a fixed FEM scenario, predictions were generated for multiple patient ages, allowing for the assessment of age-related mechanical changes in tissue stiffness as captured by the model.

The entire process is encapsulated in the pseudocode shown in Algorithm 3, which reflects the struc-

ture and automation level of the implemented codebase in Python.

Algorithm 3 Regression Analysis and Plotting Pipeline

RunRegressionPipeline Set input/output paths and load dataset Initialize regression model list **for** *each task ∈ {displacement, time, SSE}* **do**-_

Select features/target, clean data, split train/test Standardize features

for *each model* **do**

_

Train model and evaluate (R^2, MAE, RMSE, time) Rank models by R^2 and save comparison plot

if *task = displacement* **then**

_

each model with $R^2 > 0.99$ Retrain model on full data

Predict displacement for fixed age across scenarios Save scenario comparison plot

Predict displacement for fixed scenario across ages Save age effect plot

Print completion message

2.8 Results Visualization

To ensure a comprehensive and interpretable presentation of model performance, the results of the regression experiments were visualized using multiple formats tailored to different levels of analysis:

- **Performance Tables:** For each of the three regression tasks (simulated displacement, processing time, and SSE error), detailed tables were generated listing all tested models alongside their evaluation metrics: coefficient of determination (R^2), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and training time (in seconds). The tables were sorted in descending order of R^2, highlighting the most accurate models in each task.
- **Bar Charts of R^2 Scores:** To facilitate quick visual comparisons across models, horizontal bar charts were generated for each task. Each chart displays the R^2 score obtained by every model, using a consistent color palette. These plots provide an immediate overview of the relative performance and help identify clusters of well-performing regressors.
- **Line Plots of Displacement Predictions (Task 1 only):** For the displacement prediction task—deemed the most clinically relevant—additional qualitative analyses were performed. Specifically, for all models achieving $R^2 > 0.99$, the predicted displacement-versus-pressure curves were generated under varying conditions:
  - **Scenario-based Curves:** Displacement predictions were plotted for a fixed patient age across different FEM mesh configurations (scenarios). These curves were overlaid with the target displacement curve, enabling direct comparison with the results from Figures 8 and 9 of [1].
  - **Age-based Curves:** Using a fixed FEM scenario, displacement predictions were generated for patients aged 50 to 90. These curves simulate the mechanical stiffening effect of aging on corneal response, and were compared visually with the trend observed in Figure 11 of the reference paper.

All visualizations were implemented using the `matplotlib` [17] and `seaborn` [18] libraries, ensuring consistency in aesthetics, axis labeling, and color schemes. Captions, titles, and figure legends were written in English to maintain compatibility with the international scientific community. The resulting plots were automatically saved to the designated output directory (e.g., on Google Drive), facilitating integration into reports, presentations, and future publications.

3 Results

The systematic evaluation of 19 regression algorithms across the three defined prediction tasks yielded detailed quantitative and qualitative insights. The results are presented separately for each task, starting with the primary objective of simulating corneal displacement.

3.1 Prediction of Simulated Corneal Displacement (`Simulated_Displacement_mm`)

This task is the most critical, as it directly aims to emulate the behavior of finite element simulations through surrogate machine learning models. Table 1 presents a ranked summary of model performance based on the R^2 metric, while Figure 1 offers a visual comparison across all evaluated models.

Table 1: Regression Results for Target: `Simulated_Displacement_mm` (Sorted by R^2 Score)

Model	R^2 Score	MAE	RMSE	Time (s)
Extra Trees Regressor	0.9999	0.0005	0.0007	1.3602
LightGBM Regressor	0.9999	0.0007	0.0008	0.0988
Random Forest Regressor	0.9999	0.0006	0.0009	3.0264
XGBoost Regressor	0.9998	0.0009	0.0011	0.0668
Gradient Boosting Regressor	0.9997	0.0011	0.0014	0.8989
Decision Tree Regressor	0.9997	0.0012	0.0015	0.0379
MLP Regressor	0.9978	0.0030	0.0038	1.9046
KNeighbors Regressor	0.9974	0.0027	0.0041	0.0190
AdaBoost Regressor	0.9896	0.0068	0.0083	0.6520
Ridge	0.9313	0.0178	0.0214	0.0032
Bayesian Ridge	0.9313	0.0178	0.0214	0.0046
Linear Regression	0.9313	0.0178	0.0214	0.0036
Huber Regressor	0.9252	0.0170	0.0223	0.0408
Orthogonal Matching Pursuit	0.9079	0.0202	0.0247	0.0036
Passive Aggressive Regressor	0.7599	0.0329	0.0399	0.0080
SVR (Linear)	0.5182	0.0511	0.0566	0.0099
SVR (RBF)	0.3956	0.0587	0.0634	0.0199
Lasso	-0.0007	0.0679	0.0815	0.0037
ElasticNet	-0.0007	0.0679	0.0815	0.0033

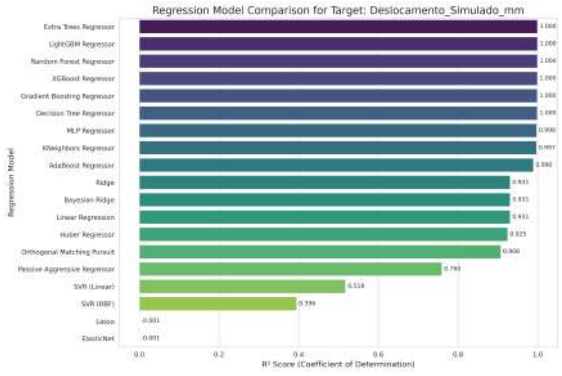

Figure 1: Comparison of R^2 scores across all models for predicting `Simulated_Displacement_mm`.

Quantitative Analysis. The analysis reveals superior performance of tree-based ensemble methods. Six algorithms achieved near-perfect prediction accuracy:

- `ExtraTreesRegressor`
- `LightGBM`
- `RandomForestRegressor`
- `XGBoost`
- `GradientBoostingRegressor`
- `DecisionTreeRegressor`

All attained exceptional $R^2 > 0.999$ with minimal prediction errors (10^{-3}–10^{-4} mm). This demonstrates their ability to:

- Capture complex nonlinear relationships
- Model intricate feature interactions
- Maintain high precision in displacement prediction

Other non-linear models, such as the Multi-Layer Perceptron (MLP) and K-Nearest Neighbors (KNN), also demonstrated high predictive accuracy ($R^2 > 0.997$). However, linear models (e.g., Ridge, Bayesian Ridge, and Linear Regression) showed limited predictive power, reaching R^2 values near 0.93. Penalized linear models like Lasso and ElasticNet performed poorly, yielding negative R^2, indicating a failure to capture the target behavior. Support Vector Regressors also underperformed, particularly with the RBF kernel.

Qualitative Analysis. For all high-performing models ($R^2 > 0.99$), we conducted visual inspections of predicted displacement curves to evaluate their biomechanical validity. Figures 2 and 3 showcase representative results from the `RandomForestRegressor`, illustrating:

- Performance across different FEM mesh scenarios (Figure 2)
- Age-dependent response patterns (Figure 3)

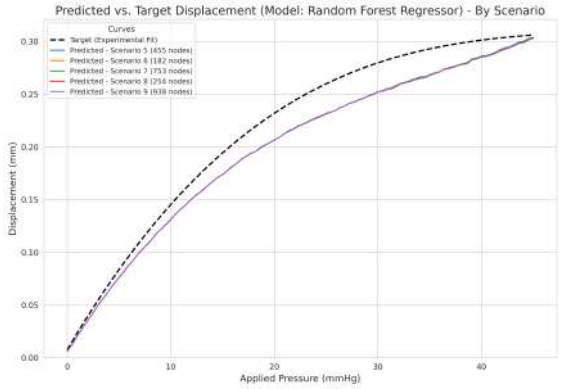

Figure 2: Predicted displacement vs. pressure for different mesh scenarios (Random Forest Regressor), compared to target curve.

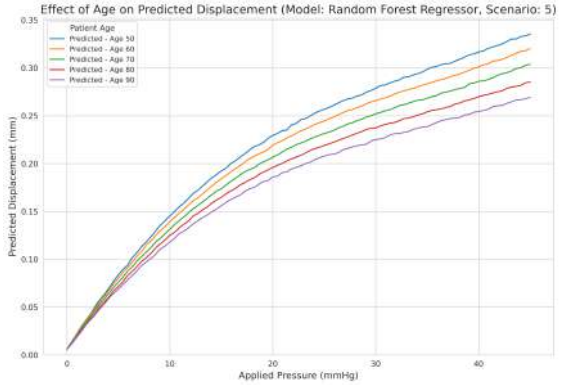

Figure 3: Predicted displacement vs. pressure at different patient ages (Random Forest Regressor, Scenario 5).

These plots provide strong visual evidence that high-performing models are not only statistically accurate but also biomechanically consistent. Specifically, they are capable of:

- Reconstructing the nonlinear sigmoidal shape of the pressure-displacement curve.
- Capturing the biomechanical stiffening effect of aging, with lower displacement for older ages.
- Differentiating the response across different FEM mesh configurations.
- Reproducing the expected deviation from the experimental target curve, as embedded in the synthetic generation logic.

The combination of numerical metrics and visual analyses confirms that several machine learning models can effectively serve as high-fidelity surrogates for computationally expensive FEM simulations in corneal biomechanics.

3.2 Prediction of Processing Time (`Processing_Time_s`)

The second regression task aimed to predict the simulation processing time, a variable strongly influenced by mesh complexity (i.e., number of nodes and elements). The target values were derived from empirical measurements reported in [1], and were fixed for each of the nine discrete scenarios in the synthetic dataset. As such, this task inherently represents a low-complexity, discrete mapping problem rather than a continuous regression challenge.

Table 2 presents the predictive performance of all evaluated models, sorted by R^2 score. A corresponding bar plot comparing R^2 values (omitted here for brevity) further illustrates the stark differences in model behavior.

As anticipated, a significant number of models achieved perfect prediction accuracy, with $R^2 = 1.0000$ and virtually zero error metrics (MAE and RMSE), including:

- Non-parametric methods: `KNeighbors Regressor`
- Tree-based models: `Decision Tree`, `Random Forest`, `Extra Trees`
- Ensemble methods: `AdaBoost`, `Gradient Boosting`, `XGBoost`, `LightGBM`

These models successfully captured the exact mapping between mesh size (number of nodes and elements) and processing time, which was constant per scenario. This result is consistent with their architectural ability to partition feature space and memorize discrete value associations, even in the presence of low data variance.

Other models, including `SVR (RBF)` and `MLP Regressor`, also achieved excellent performance ($R^2 > 0.99$), albeit with small residual errors due to their continuous approximators' nature. Linear models such as `Linear Regression`, `Ridge`, and `Bayesian Ridge` performed slightly below the best models, achieving $R^2 \approx 0.987$ — still adequate for practical purposes. The weakest results came from `ElasticNet` and `Passive Aggressive`

Table 2: Regression Results for Target: `Processing_Time_s` (Sorted by R^2 Score)

Model	R^2 Score	MAE	RMSE	Time (s)
KNeighbors Regressor	1.0000	0.0000	0.0000	0.0293
Decision Tree Regressor	1.0000	0.0000	0.0000	0.0044
Random Forest Regressor	1.0000	0.0000	0.0000	0.2774
AdaBoost Regressor	1.0000	0.0000	0.0000	0.0446
Extra Trees Regressor	1.0000	0.0000	0.0000	0.1925
XGBoost Regressor	1.0000	0.0000	0.0000	0.0373
LightGBM Regressor	1.0000	0.0001	0.0002	0.0481
Gradient Boosting Regressor	1.0000	0.0001	0.0002	0.1881
SVR (RBF)	0.9998	0.0937	0.0953	0.1204
MLP Regressor	0.9945	0.3378	0.4869	1.5435
Linear Regression	0.9870	0.6045	0.7448	0.0133
Bayesian Ridge	0.9870	0.6045	0.7448	0.0045
Ridge	0.9870	0.6043	0.7449	0.0041
Huber Regressor	0.9866	0.5958	0.7588	0.0882
SVR (Linear)	0.9842	0.5959	0.8216	3.9299
Orthogonal Matching Pursuit	0.9771	0.7398	0.9906	0.0037
Lasso	0.9620	0.9961	1.2751	0.0050
Passive Aggressive Regressor	0.9565	1.0169	1.3647	0.0089
ElasticNet	0.8800	2.1165	2.2674	0.0037

`Regressor`, although even these models achieved relatively high R^2 values (> 0.88), reinforcing the simplicity of the underlying predictive relationship.

This analysis demonstrated that when target outputs depend deterministically on categorical scenario parameters, modern machine learning models—particularly tree-based ensembles and instance-based learners—can achieve near-perfect predictions. These results highlight their exceptional suitability for:

- Low-noise regression problems
- Structure-driven predictive tasks
- Deterministic input–output relationships

3.3 Prediction of Simulation Error (`SSE_Error_Percentage`)

The third regression task focused on predicting the simulation error, expressed as the percentage of the Sum of Squared Errors (SSE) between the fitted polynomial curve and the original simulation output. This variable is particularly relevant in biomechanical modeling, as it indicates the quality of approximation in finite element post-processing [1].

Table 3 summarizes the predictive performance of all models for this task, and Figure 4 provides a visual comparison of R^2 values across the regressors.

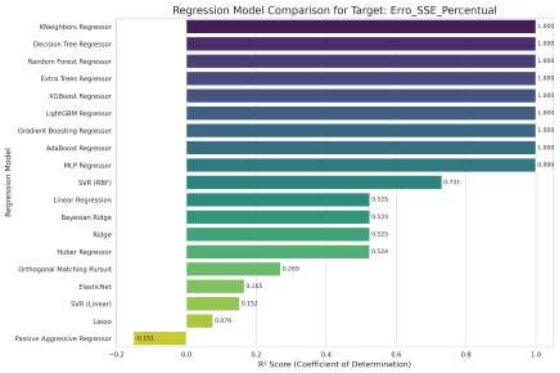

Figure 4: Performance comparison (R^2 Score) of models in predicting `SSE_Error_Percentage`.

The prediction of simulation error followed a similar pattern to the processing time task, reflecting the discrete and deterministic nature of the underlying relationship. Since the SSE percentage was also fixed per scenario in the original dataset [1], models that excel at memorizing mappings in structured spaces—such as tree-based ensembles and instance-based learners—again achieved perfect or near-perfect predictive performance.

Specifically, `KNeighbors`, `Decision Tree`, `Random Forest`, `Extra Trees`, `XGBoost`, `LightGBM`, and `Gradient Boosting` all reached $R^2 = 1.0000$ with MAE and RMSE values approach-

Table 3: Regression Results for Target: `SSE_Error_Percentage` (Sorted by R^2 Score)

Model	R^2 Score	MAE	RMSE	Time (s)
KNeighbors Regressor	1.0000	0.0000	0.0000	0.0509
Decision Tree Regressor	1.0000	0.0000	0.0000	0.0040
Random Forest Regressor	1.0000	0.0000	0.0000	0.2676
Extra Trees Regressor	1.0000	0.0000	0.0000	0.2119
XGBoost Regressor	1.0000	0.0000	0.0000	0.0409
LightGBM Regressor	1.0000	0.0001	0.0001	0.0509
Gradient Boosting Regressor	1.0000	0.0001	0.0001	0.1888
AdaBoost Regressor	0.9999	0.0099	0.0209	0.0588
MLP Regressor	0.9992	0.0402	0.0612	9.2681
SVR (RBF)	0.7310	0.6135	1.1580	1.7855
Linear Regression	0.5247	1.4134	1.5392	0.0090
Bayesian Ridge	0.5247	1.4133	1.5392	0.0045
Ridge	0.5247	1.4133	1.5392	0.0047
Huber Regressor	0.5244	1.4115	1.5397	0.0228
Orthogonal Matching Pursuit	0.2692	1.7559	1.9085	0.0037
ElasticNet	0.1655	1.9846	2.0395	0.0054
SVR (Linear)	0.1520	1.3049	2.0559	2.7016
Lasso	0.0759	2.1037	2.1462	0.0095
Passive Aggressive Regressor	-0.1512	1.8927	2.3954	0.0100

ing zero, showing they could completely reconstruct the SSE values from mesh-related features. The `AdaBoost` and `MLP Regressors` also performed remarkably well, with $R^2 > 0.999$ and small error metrics.

In contrast, linear models like `Linear Regression`, `Ridge`, and `Bayesian Ridge` struggled to capture the nonlinear associations embedded in the scenario-based patterns, yielding moderate results with $R^2 \approx 0.52$. More limited or sparsity-enforcing methods such as `Lasso`, `ElasticNet`, and `Passive Aggressive Regressor` exhibited poor performance, with R^2 dropping below 0.2 or even becoming negative, indicating that their predictions were worse than simply using the mean of the training set.

These findings further reinforce the conclusion that tasks involving discrete, scenario-dependent outputs benefit substantially from models that can exploit nonlinearity, hierarchical decision boundaries, or local similarity metrics. Moreover, they highlight the potential of tree ensembles and neighborhood-based methods in surrogate modeling where input features map deterministically to known outputs.

3.4 Comparative Analysis Across Regression Tasks

The three regression tasks addressed in this study—prediction of corneal displacement, processing time, and simulation error—presented different levels of complexity and behavior from a machine learning standpoint. A comparative evaluation of the results reveals meaningful insights into model suitability and task-specific considerations.

3.4.1 Task Complexity and Predictive Difficulty

The prediction of `Simulated_Displacement_mm` posed the greatest modeling challenge among the three tasks. Unlike the other targets, the displacement output exhibits a continuous and nonlinear dependency on multiple input features (e.g., pressure, age, mesh characteristics). As a result, only advanced models—particularly tree-based ensembles like `Extra Trees`, `LightGBM`, and `Random Forest`, as well as nonlinear learners like `MLP` and `KNN`—were able to reach R^2 values above 0.99. Linear models underperformed significantly on this task, demonstrating their limitations in capturing complex biomechanical patterns.

In contrast, the `Processing_Time_s` and `SSE_Error_Percentage` variables were deterministic and scenario-dependent. Each unique mesh configuration corresponded to a fixed processing time and

a fixed SSE% error. This enabled models that rely on discrete mapping (e.g., `Decision Tree`, `KNN`, and ensemble methods) to achieve perfect predictive performance ($R^2 = 1.0000$). Even linear models attained high R^2 scores in these tasks (e.g., $R^2 \approx 0.987$ for processing time), though with limitations for the SSE prediction.

3.4.2 Model Performance Patterns

Across all three tasks, tree-based ensemble models consistently demonstrated strong performance. These models are well-suited to heterogeneous and nonlinear data distributions, as they partition the feature space in a hierarchical, data-driven manner. They also tend to be robust to noise and require minimal preprocessing.

The `KNeighbors Regressor` also performed exceptionally in the two simpler tasks, capitalizing on the structured and repeated patterns in the synthetic dataset. However, its performance on the displacement prediction task was slightly behind that of the best ensemble models, indicating a lower generalization capacity in high-dimensional, continuous response spaces.

Linear and sparse models, such as `Lasso`, `ElasticNet`, and `Passive Aggressive Regressor`, consistently lagged across all tasks—especially for the SSE error—highlighting their inability to model nonlinear or step-like functions without prior feature engineering.

3.4.3 Implications for Surrogate Modeling

These results suggest that surrogate models based on machine learning are highly viable tools for replacing or accelerating FEM simulations, provided that model selection matches the complexity of the task. Specifically:

- For surrogate modeling of biomechanical outputs like corneal displacement, ensemble regressors and neural networks are the most effective due to their flexibility and nonlinear capacity.
- For auxiliary targets such as processing time or post-processing error, simpler models suffice, and high performance can be achieved with minimal training overhead.
- Model interpretability and inference speed may also guide model selection in real-time clinical applications.

In conclusion, the study demonstrates that the combination of synthetic datasets and diverse machine learning regressors enables robust and efficient emulation of computational biomechanics outputs, opening pathways for real-time decision support and large-scale simulations.

3.5 Visual Validation of Displacement Predictions by Top-Performing Models

After quantitative evaluation using metrics like R^2, MAE, and RMSE, a visual validation was performed for the models that demonstrated superior performance ($R^2 > 0.99$) in predicting the simulated corneal displacement. As indicated in Table 1, the models selected for this analysis were: Extra Trees Regressor, LightGBM Regressor, Random Forest Regressor, XGBoost Regressor, Gradient Boosting Regressor, Decision Tree Regressor, MLP Regressor, and KNeighbors Regressor. This qualitative analysis aims to verify whether the predictions of these models adequately capture the key functional trends present in the synthetic data, such as the non-linear pressure-displacement relationship, the effect of age, and variations between scenarios, by visually comparing them with the adjusted experimental target curve.

For each of these high-performing models, two types of line plots were generated:

1. **Comparison by Scenario:** The predicted displacement vs. pressure curves were plotted for scenarios 5 to 9 (keeping the patient's age fixed at the average of the synthetic data), overlaid on the adjusted experimental target curve. This allows for an assessment of how the model differentiates scenarios of different complexities and how its predictions compare to the experimental reference.

2. **Age Effect:** The predicted displacement vs. pressure curves were plotted for different ages (50, 60, 70, 80, 90 years), keeping the simulation scenario fixed (Scenario 5, which had the lowest SSE error in the original study [1]). This allows for an assessment of whether the model correctly captured the trend of increasing stiffness (lower displacement) with age, as incorporated into the synthetic data.

Figures 5 through 12 present the results of this visual validation for all eight selected models.

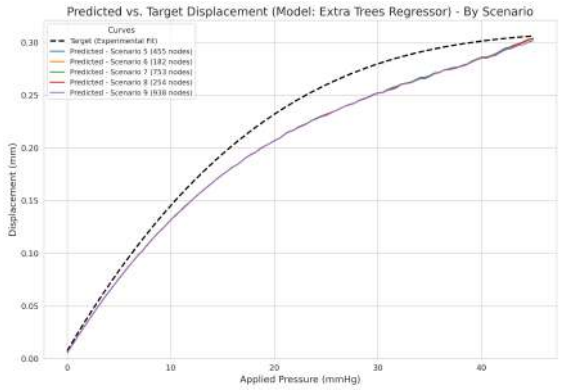

Figure 5: Comparison of predicted displacement by scenario (**Extra Trees Regressor**) vs. Target.

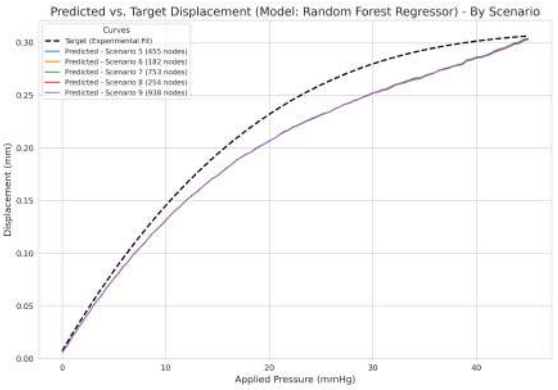

Figure 8: Comparison of predicted displacement by scenario (**Random Forest Regressor**) vs. Target.

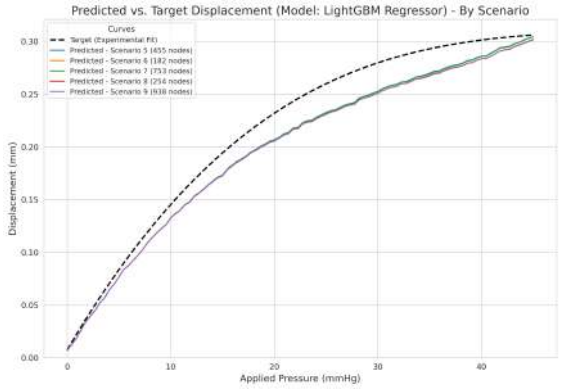

Figure 6: Comparison of predicted displacement by scenario (**LightGBM Regressor**) vs. Target.

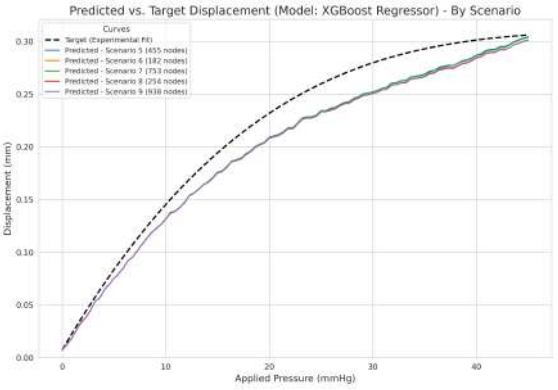

Figure 9: Comparison of predicted displacement by scenario (**XGBoost Regressor**) vs. Target.

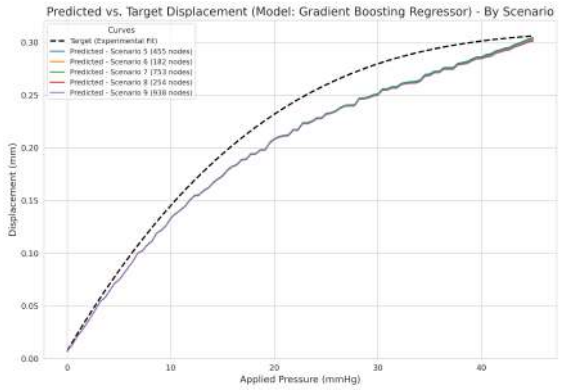

Figure 7: Comparison of predicted displacement by scenario (**Gradient Boosting Regressor**) vs. Target.

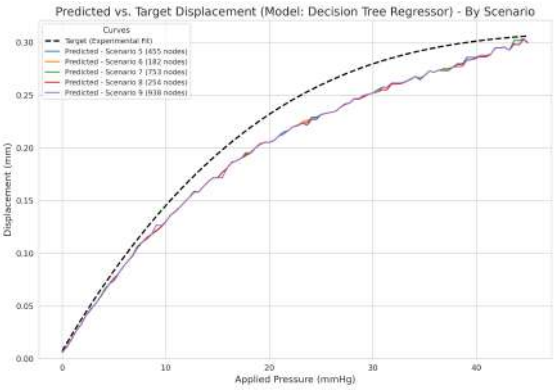

Figure 10: Comparison of predicted displacement by scenario (**Decision Tree Regressor**) vs. Target.

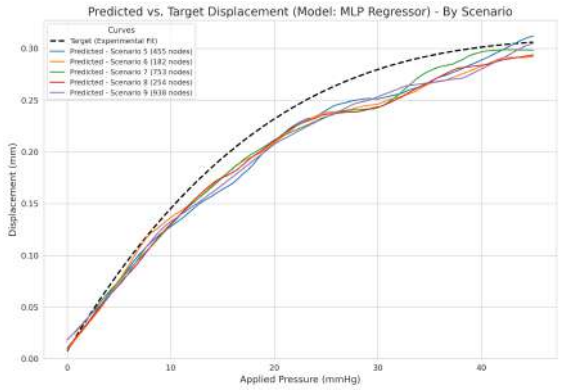

Figure 11: Comparison of predicted displacement by scenario (**MLP Regressor**) vs. Target.

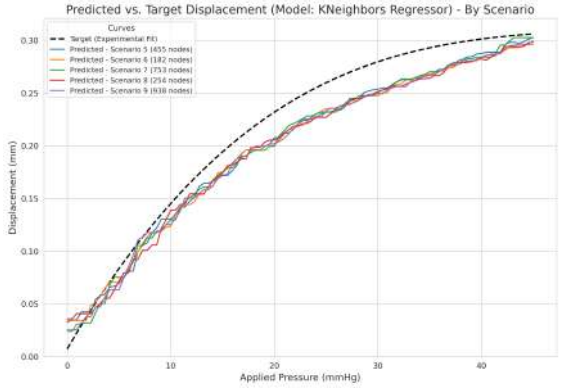

Figure 12: Comparison of predicted displacement by scenario (**KNeighbors Regressor**) vs. Target.

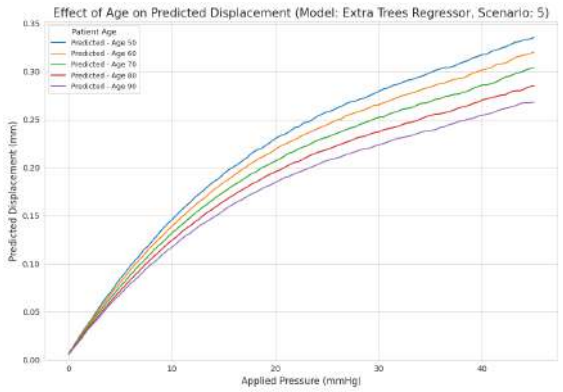

Figure 13: Effect of age on predicted displacement (**Extra Trees Regressor**, Scenario 5).

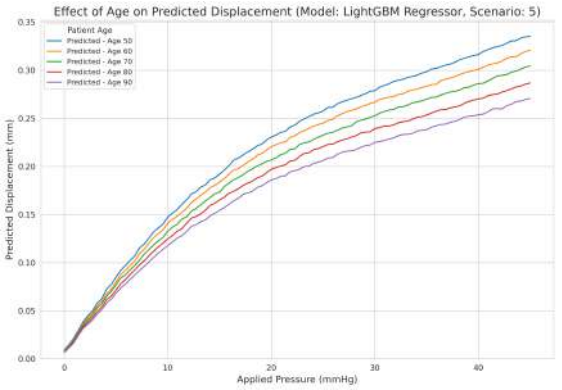

Figure 14: Effect of age on predicted displacement (**LightGBM Regressor**, Scenario 5).

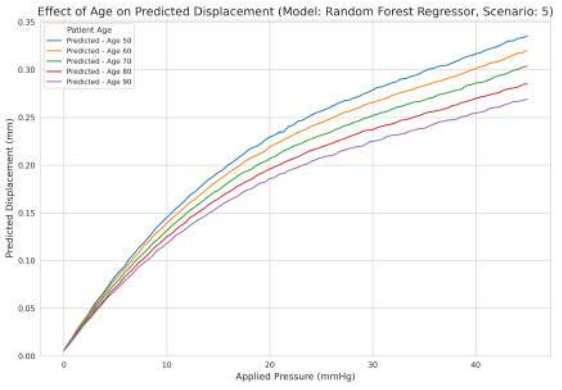

Figure 15: Effect of age on predicted displacement (**Random Forest Regressor**, Scenario 5).

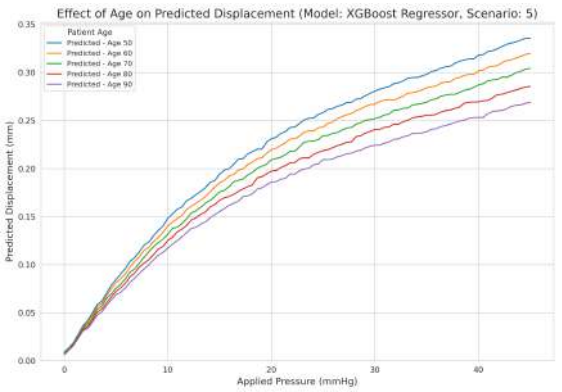

Figure 16: Effect of age on predicted displacement (**XGBoost Regressor**, Scenario 5).

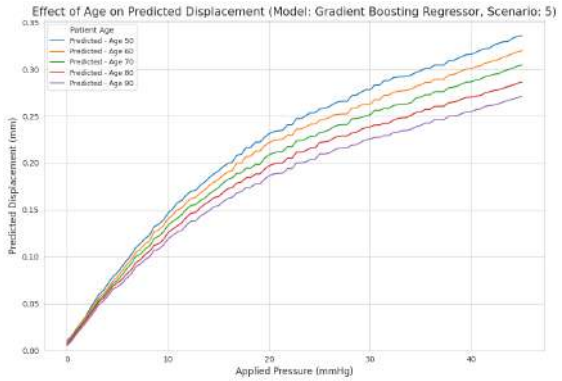

Figure 17: Effect of age on predicted displacement (**Gradient Boosting Regressor**, Scenario 5).

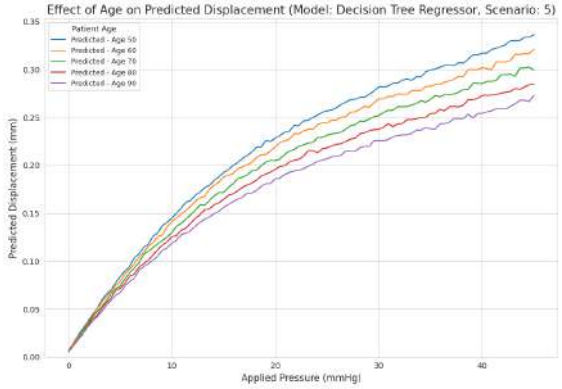

Figure 18: Effect of age on predicted displacement (**Decision Tree Regressor**, Scenario 5).

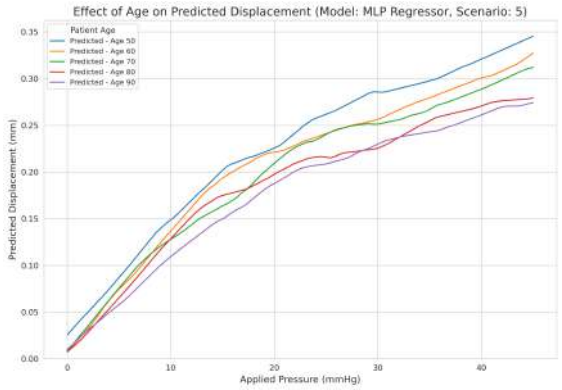

Figure 19: Effect of age on predicted displacement (**MLP Regressor**, Scenario 5).

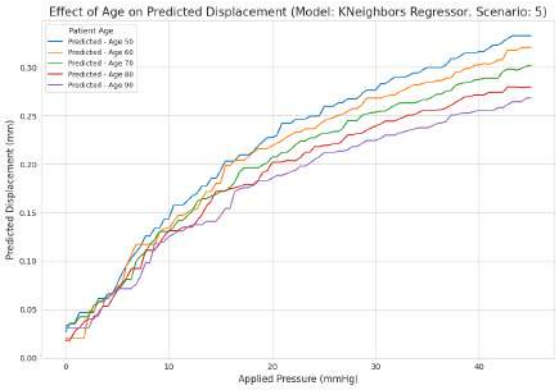

Figure 20: Effect of age on predicted displacement (**KNeighbors Regressor**, Scenario 5).

From the joint visual analysis of Figures 5 through 12, it is observed that all eight high-performing models were able to capture the non-linear pressure-displacement relationship and the modulating effect of age present in the synthetic data. Consistently, the predicted displacement curves lie below the adjusted experimental target curve, reflecting the inherent error of the original simulations that was incorporated into the synthetic data and learned by the models.

However, differences in the visual quality of the generated curves are noticeable. The tree-based ensemble models (Extra Trees, LightGBM, RandomForest, XGBoost, Gradient Boosting Regressor) produced remarkably smooth and stable predictions across the entire pressure range. The MLP Regressor also generated smooth curves, albeit with some minor ripples in certain regions. The Decision Tree Regressor, despite its high R^2, showed a slightly less smooth tendency, characteristic of single trees, and with few ripples, much less than the MLP Regressor. Finally, the KNeighbors Regressor produced visibly more irregular or "step-like" curves, an artifact of its neighbor-based methodology.

This visual validation, therefore, complements the quantitative metrics, suggesting that while all selected models are highly accurate on average (high R^2, low RMSE), the tree-based ensembles offer the most desirable combination of accuracy and smoothness in predictions, making them the most promising candidates for use as robust surrogate models in this biomechanical context.

4 Discussion

The presented results clearly demonstrate the effectiveness of the proposed methodology, which combines synthetic data generation with a comparative evaluation of Machine Learning (ML) algorithms, applied to mod-

eling aspects of corneal biomechanical behavior simulated by the Finite Element Method (FEM).

The ability of the best models—especially the tree ensembles, MLP, and KNN—to predict the `Simulated_Displacement_mm` with extremely high accuracy ($R^2 > 0.999$; RMSE < 0.001 mm) is particularly noteworthy. These results strongly validate the potential of ML algorithms as fast and accurate surrogate models for FEM simulations, at least within the parameter space covered by the synthetic data. The performance discrepancy between non-linear models and linear models ($R^2 \approx 0.93$) confirms the presence of complex, non-linear relationships between pressure, age, mesh characteristics, and displacement—as expected in biomechanical systems [7].

Figures 5 to 12 provide an important visual validation that complements the quantitative metrics. All high-performing models were able to adequately capture the non-linear behavior of the pressure-displacement curves, as well as the modulating effect of age. However, relevant differences in the smoothness of the predicted curves are observed.

The tree-based ensemble models (like Extra Trees, LightGBM, Random Forest, XGBoost, and Gradient Boosting Regressor) produced smooth and visually stable predictions across the entire pressure range (Figures 13 to 20). The MLP Regressor, despite its high R^2, exhibited occasional oscillations in some regions of the curves, suggesting that—despite its fitting capability—the model may be sensitive to the local distribution of training data. This is a typical effect for neural networks when trained with moderately small datasets or without proper regularization.

The case of the KNeighbors Regressor is even more emblematic. Despite achieving very high R^2 values, the generated curves were visibly irregular, with step-like patterns and discontinuities. This oscillation stems from the very nature of the algorithm, which makes predictions based on the average of the nearest neighbors, without continuous generalization between points. This limits its ability to generate smoothly interpolated curves, which, although not detrimental to aggregate metrics like RMSE, directly affects the interpretability and visual reliability of the results in sensitive applications such as biomechanics.

This analysis reinforces the importance of not relying solely on global performance metrics for selecting surrogate models. The visual assessment of curves is fundamental when it comes to predicting continuous physical quantities with direct clinical or biomechanical interpretation. Models with less smoothness, even with a high R^2, can lead to erroneous conclusions about the behavior of the modeled system, especially in sensitivity or optimization analyses.

In the tasks of predicting `Processing_Time_s` and `SSE_Error_Percentage`, the near-perfect performance (R^2 = 1.0) achieved by models like KNN and trees reflects the deterministic nature of the synthetic data for these variables. Since these values were fixed per scenario, the algorithms simply memorized the patterns, reproducing them with precision. Nevertheless, this behavior highlights a limitation of the representation: in practice, these variables may exhibit variations not modeled by the current approach. Still, the comparison between algorithms highlighted the difficulty of linear models in capturing these categorical relationships.

The synthetic data generation strategy was essential for the feasibility of the study. Without it, training and comparing 19 models would have required approximately 10,000 complete FEM simulations—a prohibitive cost in terms of time and computational resources. The generation process was carefully designed to preserve the functional relationships observed in the reference study [1], including the pressure-displacement curve, the effect of age, and performance metrics (time and error). The high accuracy of the trained models suggests that this approach successfully maintained these relationships. Notably, data variability was introduced only through input sampling; displacement outputs were calculated deterministically, without added random noise.

This work therefore validates a robust methodological workflow:

1. Extract knowledge from prior simulations or experiments;
2. Generate a synthetic dataset preserving this knowledge;
3. Train and select ML algorithms appropriate for the predictive task.

The resulting models—particularly tree-based ensembles—demonstrate both strong predictive performance and desirable smoothness. These models can serve as efficient tools for parametric analysis, optimization, or clinical decision support systems, provided they undergo validation with independent datasets.

5 Conclusion

This study provides compelling evidence that the integration of synthetic data generation with Machine

Learning (ML) techniques offers a powerful and efficient alternative to traditional Finite Element Method (FEM) simulations in biomechanical modeling. Focusing on the corneal response under pressure, a complex and highly nonlinear system, we demonstrated how domain-specific knowledge—extracted from a prior FEM-based study [1]—can be leveraged to build a rich and physiologically meaningful synthetic dataset comprising 10,000 samples. This dataset preserved key biomechanical phenomena, including:

- the nonlinear relationship between intraocular pressure and corneal displacement;
- the progressive stiffening of corneal tissue with increasing patient age.

Using this dataset, 19 regression algorithms were evaluated across three distinct prediction tasks: simulated displacement, processing time, and simulation error. The main findings can be summarized as follows:

- **Tree-based ensemble models** (Extra Trees, Random Forest, LightGBM, and XGBoost) emerged as top performers in the main regression task, achieving near-perfect predictive accuracy ($R^2 > 0.999$, RMSE < 0.001). In addition to their outstanding numerical performance, these models generated smooth and physiologically coherent displacement curves, making them ideal candidates for downstream clinical applications.
- **Instance-based and rule-based models** (e.g., KNeighbors and Decision Tree Regressors) exhibited excellent performance in predicting categorical-like outputs such as processing time and SSE% error—both of which were deterministic functions of the simulation mesh. Despite their limitations in producing smooth curves, these models effectively captured discrete, deterministic patterns embedded in the data.
- **Visual validation** played a pivotal role in model assessment. Although several models achieved similar numerical metrics, their qualitative behavior varied. The visual analysis revealed subtle but important differences in the smoothness, realism, and consistency of the predicted curves, emphasizing the necessity of combining quantitative and qualitative evaluations when selecting ML models for scientific and engineering applications.

Beyond the predictive performance, this work demonstrates the strategic value of synthetic data generation for:

- substantially reducing the computational burden of high-fidelity FEM simulations;
- enabling the efficient training of data-hungry ML algorithms, even in domains where collecting large real-world datasets is impractical or costly;
- establishing a replicable and generalizable framework that can be adapted to other areas of computational biomechanics, medical device modeling, or structural engineering.

The high-performing models identified in this study hold significant potential for future applications, including:

- **sensitivity analyses**, to explore the influence of multiple input parameters on corneal biomechanics;
- **parametric optimization**, such as identifying patient-specific treatment strategies or device configurations;
- **integration into real-time clinical decision support systems**, where computational speed, accuracy, and interpretability are paramount.

Nevertheless, we emphasize that the practical deployment of such ML-based surrogate models in clinical or industrial settings demands further validation. Specifically, future work should include testing on independent datasets—either from additional FEM simulations under different assumptions or, ideally, from experimental measurements (*in vitro* or *in vivo*). Only through such external validation can generalization and reliability be assured in real-world scenarios.

In conclusion, the methodological pipeline presented here showcases a promising direction for combining synthetic data and machine learning in the service of biomechanical modeling. By bridging physics-based simulation and data-driven learning, this approach paves the way for more scalable, interpretable, and clinically impactful computational tools in biomedical engineering.

References

[1] P. R. de Almeida and R. R. Magalhães, "Study of processing time for mechanical behavior analysis of corneas via numerical simulations," *Revista Brasileira de Computação Aplicada*, vol. 9, no. 4, pp. 32–42, Dec. 2017. doi: 10.5335/rbca.v9i4.7025. Available: http://seer.upf.br/index.php/rbca/article/view/7025

[2] G. L. Dias, R. R. Magalhães, F. A. Vitoriano, and D. D. Ferreira, "The use of a robotic arm for displacement measurements in a cantilever beam," *International Journal of Manufacturing, Materials, and Mechanical Engineering*, vol. 6, no. 4, pp. 45–57, 2016. doi: 10.4018/ijmmme.2016100104.

[3] C. Boote, S. Dennis, A. J. Quantock, and K. M. Meek, "Lamellar orientation in human cornea in relation to mechanical properties," *Journal of Structural Biology*, vol. 149, no. 1, pp. 1–6, Jan. 2005. doi: 10.1016/j.jsb.2004.08.009.

[4] Z. Han et al., "Biomechanical and refractive behaviors of keratoconic cornea based on three-dimensional anisotropic hyperelastic models," *Journal of Refractive Surgery*, vol. 29, no. 4, pp. 282–290, Apr. 2013. doi: 10.3928/1081597X-20130318-08.

[5] K. E. Hamilton and D. C. Pye, "Young's modulus in normal corneas and the effect on applanation tonometry," *Optometry and Vision Science*, vol. 85, no. 6, pp. 445–450, Jun. 2008. doi: 10.1097/OPX.0b013e3181783a70.

[6] A. S. Roy, M. Dupps, and W. J. Dupps Jr., "Air-puff associated quantification of non-linear biomechanical properties of the human cornea in vivo," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 48, pp. 173–182, Aug. 2015. doi: 10.1016/j.jmbbm.2015.04.010.

[7] A. Elsheikh, "Finite element modeling of corneal biomechanical behavior," *Journal of Refractive Surgery*, vol. 26, no. 4, pp. 289–300, Apr. 2010. doi: 10.3928/1081597X-20090710-01.

[8] M.Á. Ariza-Gracia, J. F. Zurita, D. P. Piñero, J. F. Rodriguez-Matas, and B. Calvo, "Coupled biomechanical response of the cornea assessed by non-contact tonometry. A simulation study," *PLoS ONE*, vol. 12, no. 3, p. e0121486, Mar. 2017. doi: 10.1371/journal.pone.0121486.

[9] I. Farmaga, M. Karpinski, and V. Tchoumatchenko, "Evaluation of computational complexity of finite element analysis," in *Proc. 11th Int. Conf. CAD Systems for Microelectronics (CADSM)*, Feb. 2011, pp. 213–214. Available: IEEE Xplore Digital Library.

[10] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," *Sensors*, vol. 18, no. 8, p. 2674, Aug. 2018. doi: 10.3390/s18082674.

[11] S. I. Nikolenko, *Synthetic Data for Deep Learning*. Cham, Switzerland: Springer, 2021. doi: 10.1007/978-3-030-75178-4.

[12] W. McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference (SciPy 2010)*, S. van der Walt and J. Millman, Eds., 2010, pp. 56–61. doi: 10.25080/Majora-92bf1922-00a.

[13] C. R. Harris et al., "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. doi: 10.1038/s41586-020-2649-2.

[14] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, Oct. 2011. Available: http://jmlr.org/papers/v12/pedregosa11a.html

[15] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[16] G. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, I. Guyon et al., Eds., 2017, pp. 3146–3154. Available: https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html

[17] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in Science Engineering*, vol. 9, no. 3, pp. 90–95, May-Jun 2007. doi: 10.1109/MCSE.2007.55.

[18] M. L. Waskom, "seaborn: statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, Apr. 2021. doi: 10.21105/joss.03021.