



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Obtención de parámetros físicos mediante aprendizaje
automático y algoritmos genéticos.

Trabajo Fin de Grado

Grado en Ciencia de Datos

AUTOR/A: Oliver Belando, Daniel

Tutor/a: Monserrat Aranda, Carlos

Cotutor/a: Rupérez Moreno, María José

CURSO ACADÉMICO: 2022/2023

RESUMEN

Este trabajo de investigación presenta un nuevo método para calcular el módulo de elasticidad de la próstata utilizando algoritmos genéticos. El módulo de elasticidad de la próstata es un parámetro que caracteriza su comportamiento biomecánico y, por tanto, determina el desplazamiento de la próstata cuando se aplica una fuerza sobre ella. Al predecir este parámetro, se puede mejorar la precisión de los modelos de simulación de la próstata.

El algoritmo genético propuesto se basa en la idea de usar una población de individuos, cada uno de los cuales representa una posible solución al problema. Los individuos se seleccionan de la población de acuerdo con su fitness, que es una medida de su calidad. Los individuos seleccionados se reproducen entre sí, y los descendientes se someten a una mutación. Este proceso se repite hasta que se encuentra una solución que satisfaga los criterios especificados.

Los resultados de este trabajo muestran que el algoritmo genético es capaz de estimar el módulo de elasticidad de la próstata con una precisión comparable a los métodos tradicionales. El algoritmo es también más rápido y eficiente que los métodos tradicionales.

Este trabajo presenta una nueva forma de estimar el módulo de elasticidad de la próstata. El algoritmo propuesto es rápido, eficiente y preciso. El algoritmo puede utilizarse para mejorar la precisión de los modelos de simulación de la próstata.

Palabras clave: Aprendizaje automático, Algoritmo genético, Módulo de elasticidad, Simulación biomecánica, Parámetros biomecánicos.

RESUM

Aquest treball de recerca presenta un nou mètode per calcular el mòdul d'elasticitat de la pròstata utilitzant algorismes genètics. El mòdul d'elasticitat de la pròstata és un paràmetre que caracteritza el seu comportament biomecànic i, per tant, determina el desplaçament de la pròstata quan s'aplica una força sobre ella. En predir aquest paràmetre, es pot millorar la precisió dels models de simulació de la pròstata.

L'algorisme genètic proposat es basa en la idea d'utilitzar una població d'individus, cadascun dels quals representa una possible solució al problema. Els individus es seleccionen de la població depenent de la seva aptitud, que és una mesura de la seva qualitat. Els individus seleccionats es reproduïxen entre ells i els descendents es sotmeten a una mutació. Aquest procés es repeteix fins que s'ha trobat una solució que compleix els criteris especificats.

Els resultats d'aquest treball mostren que l'algorisme genètic és capaç d'estimar el mòdul d'elasticitat de la pròstata amb una precisió comparable als mètodes tradicionals. L'algorisme també és més ràpid i eficient que els mètodes tradicionals.

Aquest treball presenta una nova manera de d'estimar el mòdul d'elasticitat de la pròstata. L'algorisme proposat és ràpid, eficient i precís. L'algorisme es pot utilitzar per millorar la precisió dels models de simulació de la pròstata.

Paraules clau: Aprenentatge automàtic, Algorisme genètic, Mòdul d'elasticitat, Simulació biomecànica, Paràmetres biomecànics.

ABSTRACT

This research work presents a new method for calculating the elasticity module using genetic algorithms. The prostate elasticity module is an important parameter that characterizes the biomechanical behavior of the prostate and determines the displacement of the prostate when a force is applied on it. By predicting this parameter, the accuracy of prostate simulation models can be improved.

The proposed genetic algorithm is based on the idea of using a population of individuals, each of which represents a possible solution to the problem. Individuals are selected from the population according to their fitness, which is a measure of their quality. The selected individuals reproduce with each other, and the offspring are subjected to mutation. This process is repeated until a solution that meets the specified criteria is found.

The results of this work show that the genetic algorithm is capable of estimating the prostate elastic module with accuracy comparable to traditional methods. The algorithm is also faster and more efficient than traditional methods.

This work presents a new way to estimate the prostate elastic module. The proposed algorithm is fast, efficient, and accurate. The algorithm can be used to improve the accuracy of prostate simulation models.

Keywords: Machine learning, Genetic algorithm, Elastic module, Biomechanical simulation, Biomechanical parameters.

ÍNDICE DE CONTENIDO

1. Introducción	9
1.1 Motivación	9
1.2 Objetivos	10
1.3 Metodología	11
1.4 Estructura de la memoria	11
2. Estado del arte	13
2.1 Método de elementos finitos (FEM)	13
2.1.1 Proceso	14
2.2 Algoritmos genéticos	15
2.2.1 Componentes, estructura y funcionamiento	15
2.2.2 Casos de uso	17
2.3 Redes neuronales	18
2.4 Métricas	19
2.4.1 Error cuadrático medio (<i>MSE</i>)	19
2.4.2 Error absoluto medio (<i>MAE</i>)	19
2.4.3 Coeficiente de determinación (R^2)	20
2.4.4 Precisión (<i>Accuracy</i>)	20
3. Análisis del problema	22
3.1 Análisis del marco legal y ético	22
3.1.1 Análisis de la protección de datos	22
3.1.2 Ética	23
3.2 Solución propuesta	24
4. Preparación y comprensión de datos	25
4.1 Origen de datos	25
4.2 Generación de datos	26
4.3 Empleo y evolución de los datos	27
5. Algoritmo genético	29
5.1 Inicialización	29
5.2 Evaluación	30
5.3 Evolución	31
6. Investigación y resultados	33
6.1 Metodología experimental	33
6.1.1 Diseño de los experimentos	33
6.1.2 Flujo general de trabajo	33
6.1.3 Detalles sobre la configuración de parámetros en el algoritmo genético	34



Obtención de parámetros físicos mediante aprendizaje automático y algoritmos genéticos.

7. Conclusiones	37
7.1 Relación del trabajo con los estudios cursados	37
7.2 Trabajo futuros	38
Bibliografía	40
Apéndice A – Algoritmo Genético	42
Apéndice B – Objetivos del Desarrollo Sostenible (ODS)	48

ÍNDICE DE FIGURAS

2.1 MÉTODO DE ELEMENTOS FINITOS SOBRE UN CORAZÓN. FUENTE [2].....	14
2.2 DEFORMACIÓN DE UNA PRÓSTATA ANTE UNA FUERZA EXTERNA.....	15
2.3. DIAGRAMA DE FLUJO ESTÁNDAR DE UN ALGORITMO GENÉTICO. FUENTE [3].....	17
2.4. COMPARATIVA ENTRE NEURONA BIOLÓGICA Y ARTIFICIAL. FUENTE [4].....	16
6.1. EVOLUCIÓN DEL MATERIAL Y EL ERROR ASOCIADO.....	34
6.2. ERROR DE PREDICCIÓN SEGÚN LA POBLACIÓN INICIAL.....	35

ÍNDICE DE ECUACIONES

2.1 ECUACIÓN MSE.....	19
2.2 ECUACIÓN MAE.....	20
2.3 ECUACIÓN R^2	20
2.4 ECUACIÓN ACCURACY.....	20
5.1 ECUACIÓN COMBINATORIA SIN REPETICIÓN.....	31



Obtención de parámetros físicos mediante aprendizaje automático y algoritmos genéticos.

1. INTRODUCCIÓN

El aprendizaje automático (*Machine Learning*, ML) es una rama de la inteligencia artificial que permite a los ordenadores aprender sin ser programados explícitamente. En medicina, el ML se está utilizando para mejorar el diagnóstico, el pronóstico y el tratamiento de una amplia gama de enfermedades.

Uno de los campos en los que el ML está teniendo un impacto significativo es el de la simulación biomecánica. La simulación biomecánica es el uso de modelos informáticos para estudiar el comportamiento mecánico de los sistemas biológicos. Esto puede utilizarse para mejorar la comprensión de las enfermedades, desarrollar nuevos tratamientos y diseñar dispositivos médicos personalizados.

El aprendizaje automático puede mejorar la simulación biomecánica de varias maneras. En primer lugar, el ML puede utilizarse para generar datos más precisos. Esto se debe a que el ML puede aprender de los datos y corregir los errores que pueden estar presentes en los datos experimentales. En segundo lugar, el ML puede utilizarse para desarrollar modelos más complejos. Esto se debe a que el ML puede aprender de datos multidimensionales que pueden ser difíciles de modelar con métodos tradicionales. En tercer lugar, el ML puede utilizarse para automatizar el proceso de simulación. Esto puede liberar a los investigadores para que se concentren en tareas más creativas y estratégicas.

El impacto del ML en la simulación biomecánica está aún en sus primeras etapas, pero tiene el potencial de revolucionar la forma en que se estudia y trata las enfermedades. A medida que el ML siga desarrollándose, se espera que tenga un impacto aún mayor en el campo de la medicina.

1.1 MOTIVACIÓN

Hoy en día, la inteligencia artificial (IA) está adentrándose en cada uno de nuestros ámbitos y, sin duda alguna, su relevancia es fundamental en el campo de la medicina. La salud es un aspecto primordial en nuestras vidas y cualquier avance tecnológico que pueda contribuir a mejorarla es de suma importancia.

En este contexto, la obtención de parámetros físicos mediante aprendizaje automático y algoritmos genéticos (AG) representa un área de investigación prometedora. La combinación de estas técnicas nos brinda la oportunidad de adentrarnos en la comprensión de la anatomía y el funcionamiento del cuerpo humano de una manera más precisa y eficiente.

El cálculo del módulo de elasticidad de la próstata utilizando algoritmos genéticos se vuelve relevante en este escenario. La próstata es un órgano de vital importancia en la salud masculina y comprender sus propiedades biomecánicas es fundamental para el diagnóstico y tratamiento de diversas afecciones relacionadas. La capacidad de predecir el desplazamiento en respuesta a una fuerza aplicada en una zona específica de la próstata a través de la obtención precisa de este parámetro puede revolucionar la forma en que se planifican los tratamientos y se abordan las intervenciones quirúrgicas.

Además, el uso de AG, inspirados en los procesos evolutivos naturales, puede ayudarnos a encontrar soluciones óptimas en la obtención de los parámetros físicos



deseados aprovechando la capacidad de optimización que ofrecen, mejorando así la precisión y eficiencia de los resultados.

La motivación detrás de este trabajo radica en la pasión por explorar las posibilidades que la inteligencia artificial y los algoritmos genéticos brindan en el ámbito médico. Contribuir al avance de la ciencia y la tecnología en este campo es un objetivo que impulsará nuestro compromiso y dedicación en este proyecto. En última instancia, aspiramos a mejorar la comprensión de la próstata y su impacto en la salud masculina, con el objetivo de brindar soluciones innovadoras que beneficien a los pacientes y promuevan un enfoque más personalizado y eficaz en la medicina.

1.2 OBJETIVOS

El objetivo principal de este trabajo de fin de grado es hacer una ampliación del trabajo final de grado de Andrea García Pastor, titulado “Aprendizaje automático en la aceleración de cálculos biomecánicos: aplicación a la simulación del comportamiento de próstatas” [1].

Esta ampliación consiste en desarrollar un modelo capaz de obtener el módulo de elasticidad que modela el comportamiento biomecánico de la próstata mediante la utilización de algoritmos genéticos y aprendizaje automático. En concreto, se busca calcular el Módulo de Young¹ de la próstata, un parámetro fundamental para predecir el desplazamiento en respuesta a una fuerza aplicada en una zona específica de este órgano.

Para alcanzar este objetivo, se llevarán a cabo las siguientes tareas:

1. Realización de simulaciones utilizando el método de elementos finitos para estudiar el comportamiento biomecánico de la próstata bajo diferentes condiciones de fuerza y posición.
2. Recopilación y análisis de los datos obtenidos en las simulaciones para construir un conjunto de datos completo y representativo.
3. Preparación de los datos recopilados para su uso en el diseño del algoritmo genético.
4. Investigación y evaluación de diferentes estrategias para la evolución de la población y técnicas de aprendizaje automático que permitan predecir con precisión el módulo de elasticidad de las distintas próstatas.
5. Validación del algoritmo desarrollado mediante comparación con los parámetros físicos reales obtenidos de datos clínicos, evaluando la capacidad de predicción del algoritmo creado.

En resumen, el objetivo de este trabajo es desarrollar un algoritmo genético que permita calcular el módulo de elasticidad o Módulo de Young de la próstata a partir de distintos datos obtenidos gracias a un modelo pre-entrenado. La idea es, por tanto,

¹ Parámetro que caracteriza el comportamiento de un material elástico

predecir el coeficiente de material únicamente sabiendo cómo se desplazan los nodos en función de una fuerza ejercida sobre un nodo de la próstata. Estas predicciones pueden tener implicaciones importantes en la planificación de tratamientos médicos y en la comprensión de los mecanismos biomecánicos de este órgano.

1.3 METODOLOGÍA

Para el proyecto se han utilizado distintas herramientas y procedimientos para generar los datos, crear el algoritmo y evaluar resultados.

Las herramientas que se han utilizado son las siguientes:

- Por una parte, el software llamado Ansys Student² especializado en elementos finitos (FEM) del que obtenemos la información propia de cada una de las próstatas.
- Por otro lado, hemos utilizado Python, uno de los lenguajes de programación más utilizados hoy en día, para predecir los desplazamientos de las próstatas por el modelo pre-entrenado como para crear el algoritmo genético. Destacan las librerías *Pandas*³, *Numpy*⁴, *Matplotlib*⁵, *Tensorflow*⁶, *Keras*⁷ o *Itertools*⁸, que son las que han sido de requeridas para el desarrollo del algoritmo.

En cuanto a los procedimientos llevados a cabo, se ha tratado de hacer el proyecto lo más replicable posible, de tal forma que, si una persona ajena al trabajo quiere utilizar los datos y el algoritmo creado, pueda sin ningún tipo de dificultad. Para ello, hemos utilizado formato *Excel* o Archivos de texto para los datos, mientras que para el algoritmo se ha empleado un tipo de estructura llamada *Clase*⁹ que permite definir un conjunto de métodos y atributos relacionados a un objeto o entidad, que en este caso sería el algoritmo genético.

1.4 ESTRUCTURA DE LA MEMORIA

La memoria está compuesta por los seis capítulos que se explican a continuación:

1. **Introducción:** Se proporciona una visión general del tema, se establece su relevancia y se plantean los objetivos y la estructura del trabajo.
2. **Estado del arte:** Se presentan y analizan de manera detallada las investigaciones, avances y desarrollos más recientes relacionados con el tema de estudio, además de mostrar otros conceptos necesarios para entender el proyecto en su totalidad.
3. **Análisis del problema:** Se contextualiza la problemática que el trabajo abordará. Además, se establece la relevancia y justificación de resolver dicho problema en el contexto científico.

² <https://www.ansys.com>

³ <https://pandas.pydata.org>

⁴ <https://numpy.org>

⁵ <https://matplotlib.org>

⁶ <https://www.tensorflow.org>

⁷ <https://keras.io>

⁸ <https://docs.python.org/3/library/itertools.html>

⁹ <https://docs.python.org/es/3/tutorial/classes.html>



Obtención de parámetros físicos mediante aprendizaje automático y algoritmos genéticos.

4. **Preparación y comprensión de datos:** Se explica el proceso paso a paso desde que se obtienen los datos hasta que están en su formato final, de tal forma que se puede usar para el crear el algoritmo.
5. **Evaluación del algoritmo:** Este capítulo muestra las distintas pruebas realizadas sobre el algoritmo. Además, se exponen las decisiones tomadas sobre el funcionamiento del algoritmo.
6. **Resultados:** Se indican qué pruebas resultan ser las más eficaces. Se relacionan distintos conceptos como tiempo de computación o simplicidad del algoritmo con el rendimiento de este. También se muestran gráficas y tablas.
7. **Conclusiones:** Se hace una recopilación de los objetivos principales con tal de ver si se han cumplido, se exponen las limitaciones presentes durante el trabajo y los proyectos futuros. Además, se enlaza el proyecto desarrollado con los estudios cursados.

2. ESTADO DEL ARTE

El avance tecnológico en el campo de la biomecánica ha permitido a los investigadores explorar con mayor profundidad la interacción mecánica de los tejidos vivos. Uno de los enfoques más prometedores es la predicción del módulo de elasticidad en materiales anisótropos¹⁰. Para contextualizar y comprender el entorno en el que se inscribe este trabajo, es fundamental explorar los estudios previos, los métodos utilizados y las métricas empleadas en este ámbito de investigación.

Entre los avances mencionados, el más destacado en el ámbito de la medicina ha sido el surgimiento de la inteligencia artificial. Esta tecnología ha ampliado significativamente las posibilidades en el campo médico, dando lugar a numerosas investigaciones que aprovechan sus capacidades para optimizar procesos clínicos e incluso para realizar diagnósticos precisos de pacientes, como se evidencia en estudios previos ([8], [11], [15]).

Tanto los algoritmos genéticos como la predicción del módulo de elasticidad en materiales elásticos han sido objeto de investigaciones previas. En el caso de los algoritmos genéticos, se han llevado a cabo estudios que se enfocan en la optimización de sus parámetros y en el perfeccionamiento de su funcionamiento ([12], [13]). Además, existen investigaciones que se centran en la aplicación de estos algoritmos en contextos empresariales. ([14])

En lo que respecta a la predicción del módulo de Young, investigaciones recientes, como se detalla en ([10]), han logrado predecir este parámetro utilizando el método de elementos finitos (FEM), que se explicará en detalle en secciones posteriores.

2.1 MÉTODO DE ELEMENTOS FINITOS (FEM)

El Método de Elementos Finitos (FEM) es una técnica numérica ampliamente utilizada en ingeniería y ciencias aplicadas para resolver problemas de análisis estructural y comportamiento mecánico de materiales y objetos complejos. Es una herramienta poderosa que permite modelar y simular el comportamiento de estructuras, sólidos y fluidos bajo diferentes condiciones y cargas. Este método es ampliamente utilizado en el ámbito de la medicina, además se puede ver en estudios en los que se trata de predecir también el módulo de Young ([10]).

La esencia del Método de Elementos Finitos radica en dividir el dominio de estudio, es decir, el objeto o material que se quiere analizar, en elementos más pequeños y sencillos llamados elementos finitos. Como se puede apreciar en la figura 2.1. estos elementos se caracterizan por tener una geometría y propiedades mecánicas específicas, lo que permite simplificar el problema global en una serie de problemas locales más manejables.

¹⁰

adj. Fís. Dicho de una sustancia o de un cuerpo: Que posee propiedades físicas distintas según la dirección en que se mide. (Real Academia Española, s.f., definición 1).

Obtención de parámetros físicos mediante aprendizaje automático y algoritmos genéticos.

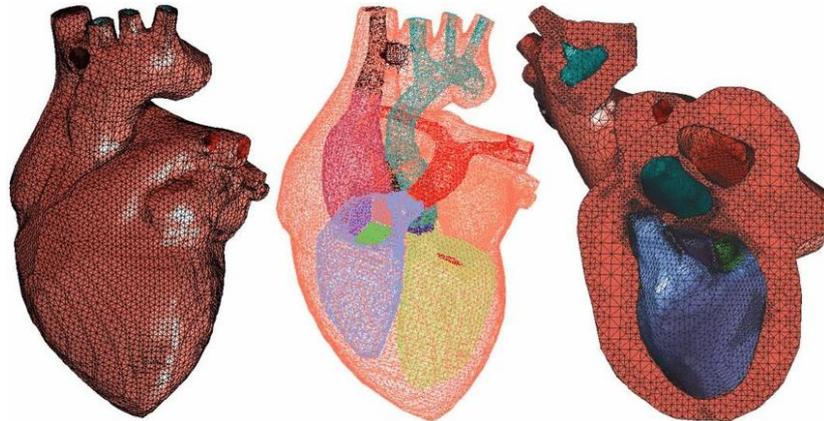


FIGURA 2.1: MÉTODO DE ELEMENTOS FINITOS SOBRE UN CORAZÓN. FUENTE [2]

2.1.1 PROCESO

El proceso de análisis mediante FEM implica los siguientes pasos:

1. **Discretización del dominio:** El objeto o material se divide en una malla de elementos finitos que representa su geometría. Estos elementos pueden ser triángulos, cuadrados, tetraedros, hexaedros, entre otros, dependiendo de la naturaleza del problema.
2. **Formulación de las ecuaciones de equilibrio:** Se establecen las ecuaciones de equilibrio y comportamiento mecánico de cada elemento finito, teniendo en cuenta las condiciones de carga, restricciones y propiedades del material.
3. **Montaje del sistema global:** Se combinan las ecuaciones locales de cada elemento para formar un sistema global de ecuaciones, que representa el comportamiento mecánico completo del objeto o material en estudio.
4. **Solución del sistema de ecuaciones:** Se resuelve el sistema global de ecuaciones mediante métodos numéricos, como el método de eliminación gaussiana, para obtener las soluciones de desplazamientos, deformaciones y tensiones en cada punto del dominio.
5. **Evaluación y post-procesamiento de resultados:** Se analizan los resultados obtenidos para comprender el comportamiento mecánico del objeto o material bajo diferentes condiciones de carga y obtener información relevante, como las deformaciones máximas, tensiones críticas, etc. Ver la figura 2.2.

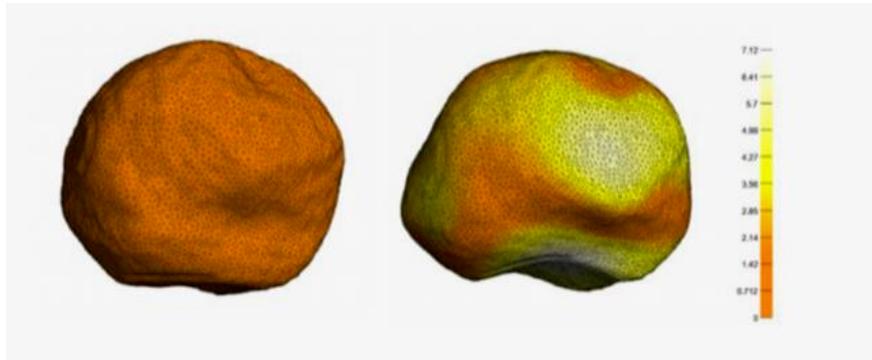


FIGURA 2.2. DEFORMACIÓN DE UNA PRÓSTATA ANTE UNA FUERZA EXTERNA.

2.2 ALGORITMOS GENÉTICOS

Los algoritmos genéticos son una familia de modelos computacionales inspirados en la evolución. Estos algoritmos pueden resolver tanto problemas de minimización como de maximización. Para ello, codifican una solución potencial a un problema específico en una estructura de datos simple similar a un cromosoma y aplican operadores de recombinación a estas estructuras para preservar la información crítica.

Estos algoritmos suelen considerarse optimizadores de funciones, aunque la gama de problemas a los que se han aplicado los algoritmos genéticos es bastante amplia. Una aplicación de algoritmo genético comienza con una población de cromosomas (normalmente aleatorios). A continuación, se evalúan estas estructuras y se asignan oportunidades de reproducción de tal forma que los cromosomas que representan una mejor solución al problema objetivo tengan más oportunidades de 'reproducirse' que aquellos cromosomas que son soluciones más pobres. La "bondad" de una solución suele definirse con respecto a la población actual.

2.2.1 COMPONENTES, ESTRUCTURA Y FUNCIONAMIENTO

En esta sección, se abordará en detalle la composición, la estructura y el funcionamiento intrínseco de los componentes clave del algoritmo propuesto. Al explorar los elementos esenciales de la metodología, se obtendrá una comprensión más profunda de cómo se lleva a cabo el proceso de optimización. A través del desarrollo de su estructura, se comprenderá mejor la capacidad del algoritmo para adaptarse y evolucionar en la búsqueda de soluciones óptimas. Ver figura 2.3.

- **Codificación:** Es el proceso en el cual representamos un problema real de forma que pueda ser solucionado haciendo uso de algoritmos genéticos. En función del problema a solucionar, la codificación estará en mayor o menor medida presente porque habrá problemas que sin ninguna gran modificación puedan ser resueltos por estos tipos de algoritmos.

Las codificaciones pueden ser de distintos tipos, pero en la literatura se resaltan tres esquemas: codificación binaria, codificación por permutación y codificación de valores.

Obtención de parámetros físicos mediante aprendizaje automático y algoritmos genéticos.

- **Inicialización:** Primer paso del algoritmo genético. El problema ya lo hemos transformado a su forma genética y creamos la población. Hay muchas técnicas distintas sobre la generación de una población inicial pero la generación aleatoria suele ser la más apropiada.
- **Selección:** Es uno de los operadores genéticos que se utiliza. Consiste en seleccionar un cierto número de individuos de la población, los cuales serán cruzados en la siguiente generación. Es decir, estos individuos pertenecerán a la siguiente población. Los individuos con mejores resultados o cualidades tienen mayor probabilidad de ser seleccionados.
- **Reproducción:** Es el segundo operador genético que se utiliza. Este es el proceso de generar hijos a partir de padres. Los individuos seleccionados por el operador "Selección" son los que harán el rol de padre. Hay distintas técnicas para llevar a cabo la reproducción, pero el objetivo sigue siendo el mismo que es generar descendientes que combinen características de los padres para que haya variabilidad en la siguiente población.
- **Mutación:** Último operador genético que se utiliza en los algoritmos genéticos. El proceso que se lleva a cabo en esta fase es la de modificar o cambiar una pequeña fracción de la población de forma aleatoria y no dependiente de poblaciones anteriores u otros individuos.
- **Evaluación:** Para cada población hay un proceso en el que se hace uso de una función matemática para calcular la calidad de la solución.

El valor medio de calidad de la población representa la como de buena es la población en relación con el objetivo esperado. Los operadores genéticos mejoran la media de la población en cada iteración y el resultado final es aquel individuo que alcance una solución óptima.

Hay distintos tipos de métricas según el tipo de penalización que se quiera realizar o cómo se quiera valorar la calidad del resultado. Esta información se explica en el apartado *2.5 Métricas*.

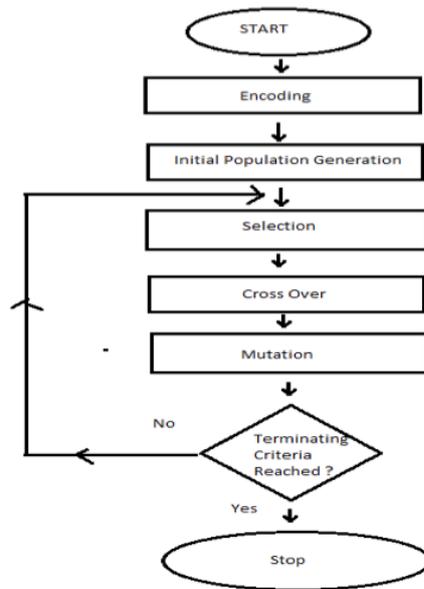


FIGURA 2.3. DIAGRAMA DE FLUJO ESTÁNDAR DE UN ALGORITMO GENÉTICO. FUENTE [3]

2.2.2 CASOS DE USO

El proceso de análisis mediante FEM implica los siguientes pasos:

Los algoritmos genéticos están presentes en numerosos campos de investigación o desarrollo. A continuación, se detallan algunas de las aplicaciones de estos algoritmos:

- **Videojuegos:** Se utiliza para optimizar problemas donde el espacio de búsqueda es inmenso. Ciertas decisiones que toma el propio videojuego son debido a algoritmos genéticos que calculan distintas situaciones para una experiencia óptima del jugador.
- **Programación de actividades:** En este tipo de problemas hay diferentes horarios, costes, beneficios e incluso, a veces, máquinas/personas con diferentes capacidades. Los algoritmos genéticos han sido útiles para encontrar soluciones óptimas a estos problemas.
- **Robótica:** De manera similar a los videojuegos, los AG se utilizan para generar distintas posibles situaciones a partir de la combinación de trayectorias o movimientos. Gracias a lo que se consigue una solución óptima para el movimiento del robot dentro del entorno en el que se encuentra.
- **Aprendizaje automático:** Los AG también son fundamentales dentro de la búsqueda de hiperparámetros¹¹ para modelos de aprendizaje automático. Estos hiperparámetros son los parámetros que no se optimizan durante los entrenamientos

¹¹ Parámetro cuyo valor se utiliza para controlar el proceso de aprendizaje. No es inferen del entrenamiento del modelo.

Obtención de parámetros físicos mediante aprendizaje automático y algoritmos genéticos.

de los modelos, sino que deben de ser elegidos por el usuario, por ejemplo, el número de capas ocultas de una red neuronal.

Estas son algunas de las numerosas aplicaciones que tienen los Algoritmos genéticos. Su relevancia es dada gracias a la facilidad para trabajar con un inmenso espacio de soluciones, posibilidad de paralelizar, no tiene grandes requerimientos computacionales, además de la propia sencillez del concepto (evolución de una población).

2.3 REDES NEURONALES

Es importante comprender qué es una red neuronal (*Neural Network*, NN) y cómo ha influido en el desarrollo de la inteligencia artificial y el aprendizaje automático, ver figura 2.4. Una red neuronal es un modelo matemático inspirado en la estructura y funcionamiento del cerebro humano. Consiste en una serie de nodos interconectados, también llamados neuronas artificiales, que procesan y transmiten información.

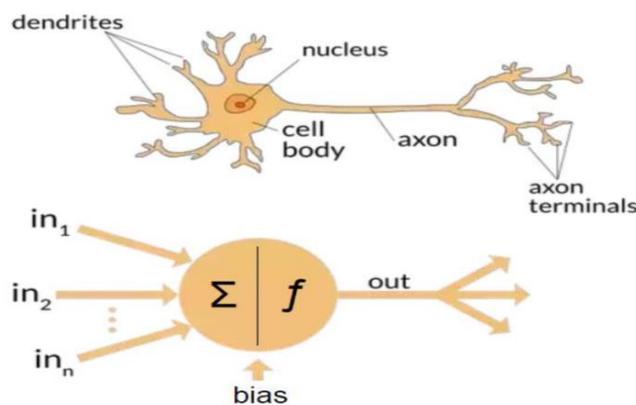


FIGURA 2.4. COMPARATIVA ENTRE NEURONA BIOLÓGICA Y ARTIFICIAL. FUENTE [4]

Cada neurona artificial recibe entradas, las procesa a través de una función de activación y produce una salida. Estas salidas se convierten en las entradas para las siguientes capas de neuronas, creando así un tejido de conexiones interconectadas. La interacción entre estas neuronas y las conexiones entre ellas es lo que permite que las redes neuronales sean capaces de aprender y adaptarse a patrones complejos en los datos.

En el contexto del aprendizaje automático, las redes neuronales son la base de muchas técnicas exitosas, como el aprendizaje profundo (*Deep Learning*). Las redes neuronales profundas son aquellas que tienen múltiples capas intermedias entre la capa de entrada y la capa de salida. Estas capas intermedias, también conocidas como capas ocultas, permiten a la red aprender representaciones jerárquicas y abstractas de los datos, lo que es esencial para el procesamiento de información compleja, como imágenes, audio o texto.

El entrenamiento de una red neuronal implica ajustar los pesos y los sesgos de las conexiones entre las neuronas para minimizar la diferencia entre las salidas predichas

y las salidas reales conocidas. Esto se logra mediante algoritmos de optimización que ajustan gradualmente los parámetros de la red en función de la retroalimentación proporcionada por los datos de entrenamiento.

En resumen, una red neuronal es un modelo computacional que emula el funcionamiento del cerebro humano y ha revolucionado el campo del aprendizaje automático al permitir a las máquinas aprender y realizar tareas complejas como reconocimiento de patrones, procesamiento de lenguaje natural y visión por computadora.

2.4 MÉTRICAS

Dentro del contexto del aprendizaje automático, es fundamental entender las métricas utilizadas para evaluar y comparar el rendimiento de estos modelos en tareas de aprendizaje automático. Estas métricas proporcionan una medida cuantitativa de la capacidad predictiva de los modelos y son esenciales para determinar su eficacia en diversas aplicaciones ([11]).

En el caso de algoritmos genéticos, se evalúa su desempeño mediante métricas que miden la calidad de las soluciones encontradas en cada generación. Una métrica común es la aptitud o "fitness", que cuantifica la calidad de una solución dada. A lo largo de las generaciones, se espera que la aptitud de las soluciones mejore a medida que el algoritmo evoluciona y selecciona soluciones más prometedoras. Además, también se pueden considerar métricas relacionadas con la diversidad de las soluciones para garantizar una exploración efectiva del espacio de búsqueda. Todo esto ya explicado en el apartado 2.3 *Algoritmos genéticos*.

En cualquier caso que se utilice o se haga uso del aprendizaje automático, se hace uso de las métricas de evaluación, aunque pueden variar entre distintos modelos o algoritmos según el tipo de tarea que se intenta optimizar o la naturaleza de los datos.

Algunas de las métricas más comunes son:

2.4.1 ERROR CUADRÁTICO MEDIO (*MSE*)

Es una métrica común en problemas de regresión. Calcula el promedio de los cuadrados de las diferencias entre las salidas predichas por el modelo y las salidas reales. Cuanto menor sea el MSE, mejor será el ajuste del modelo a los datos. Sin embargo, el MSE tiende a castigar los errores grandes de manera más significativa debido a la elevación al cuadrado, lo que puede dar lugar a modelos sensibles a valores atípicos.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

Ecuación 2.1: Ecuación MSE.

2.4.2 ERROR ABSOLUTO MEDIO (*MAE*)

También se utiliza en problemas de regresión. En lugar de elevar al cuadrado las diferencias entre las predicciones y los valores reales, toma los valores absolutos de esas diferencias y calcula su promedio. El MAE es menos sensible a valores atípicos



Obtención de parámetros físicos mediante aprendizaje automático y algoritmos genéticos.

que el MSE, lo que lo hace más robusto en presencia de datos anómalos, y puede ser más útil en situaciones donde las diferencias pequeñas son más importantes.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

Ecuación 2.2: Ecuación MAE.

2.4.3 COEFICIENTE DE DETERMINACIÓN (R^2)

Proporciona una medida de qué tan bien se ajustan las predicciones del modelo a los datos reales. Representa la proporción de la varianza en las salidas reales que es explicada por las salidas predichas. Un valor de R^2 cercano a 1 indica un ajuste muy bueno, mientras que un valor cercano a 0 sugiere que el modelo no es capaz de explicar la variabilidad en los datos.

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2}$$

Ecuación 2.3: Ecuación R^2 .

2.4.4 PRECISIÓN (*ACCURACY*)

Es una métrica clave en problemas de clasificación. Mide la proporción de instancias clasificadas correctamente en relación con el total de instancias. Es simple y fácil de entender. Sin embargo, puede ser engañosa en problemas de clasificación desequilibrados, donde una clase es mucho más frecuente que la otra.

$$Accuracy = \frac{True\ Positives + True\ Negative}{N^{\circ}\ of\ all\ samples}$$

Ecuación 2.4: Ecuación *Accuracy*.

En resumen, cada una de estas métricas tiene su propia utilidad y aplicabilidad en diferentes contextos. La elección de la métrica depende de los objetivos del análisis y el tipo de problema que se está abordando. Es importante comprender cómo se calculan y qué representan. La decisión sobre qué métrica elegir depende de consideraciones específicas de la aplicación y de qué aspecto del rendimiento del modelo es más crucial en el contexto dado.



3. ANÁLISIS DEL PROBLEMA

Tras haber examinado minuciosamente el panorama actual en el estado del arte es imperativo emprender un análisis riguroso del problema que el presente trabajo de investigación aborda. Este análisis no solo permite una comprensión más profunda de la complejidad inherente al tema, sino que también abre la puerta a la identificación de cuestiones cruciales en lo relacionado con la protección de datos y la ética del proyecto.

La conducción de este análisis se realizará de manera metódica y sistemática, haciendo uso de las técnicas y enfoques más adecuados conforme a la naturaleza intrínseca del problema a resolver. Este proceso se erige como una etapa sustancial en la búsqueda de soluciones significativas y en la delimitación de los objetivos claros que guiarán el desarrollo del presente trabajo.

3.1 ANÁLISIS DEL MARCO LEGAL Y ÉTICO

En el proceso de abordar cualquier proyecto con relevancia en el ámbito médico y científico, resulta crucial considerar las implicaciones legales y éticas asociadas. Este apartado se establece como un pilar fundamental en esta investigación, con el objetivo central de examinar de manera exhaustiva los aspectos legales y éticos que rodean el uso de datos relacionados con próstatas y la obtención de información sobre sus comportamientos físicos mediante el empleo de algoritmos genéticos.

Este análisis proveerá una base sólida para garantizar la integridad, la legalidad y la responsabilidad en todas las etapas del proyecto, respetando tanto las regulaciones vigentes como los principios éticos que rigen la investigación médica y científica.

3.1.1 ANÁLISIS DE LA PROTECCIÓN DE DATOS

En este apartado vamos a realizar un análisis minucioso de la protección de datos enfocada a nuestro proyecto.

La normativa actual con relación a la protección de datos está enmarcada en el RGPD (Reglamento General de Protección de Datos, [5]). Sin embargo, considerando la naturaleza de los datos médicos involucrados, es esencial tener en cuenta las regulaciones específicas del ámbito sanitario, como la LOPDGDD (Ley Orgánica de Protección de Datos y Garantía de Derechos Digitales, [6]) y la Ley de Autonomía del Paciente ([7]). Estas leyes establecen criterios adicionales en cuanto a la confidencialidad de los datos médicos y la salvaguardia de la información sanitaria, asegurando una protección adecuada de los derechos y la privacidad de los pacientes.

Para juzgar si nuestros datos cumplen con las normativas pertinentes, vamos a definir qué aborda la Ley de Protección de Datos Sanitarios. Esta aborda la gestión de información de salud con énfasis en la confidencialidad y privacidad de los pacientes. Establece reglas estrictas para la recopilación, almacenamiento y procesamiento de datos médicos. Los aspectos fundamentales incluyen obtener el consentimiento informado de los pacientes, garantizar el acceso restringido a los datos por parte de profesionales autorizados y adoptar medidas de seguridad sólidas para prevenir el acceso no autorizado. También se promueve la transparencia en la gestión de datos y se otorgan derechos a los pacientes para acceder, rectificar y eliminar su información médica.

Las reglas establecidas en la Ley de Protección de Datos Sanitarios son fundamentalmente respaldadas por la forma en que se obtuvieron los datos para este estudio. Los datos se originaron a partir de imágenes de resonancia magnética (IRM) de pacientes, quienes proporcionaron su consentimiento firmado para su uso con fines investigativos. Además, se implementó un proceso de anonimización de los datos para eliminar cualquier identificación personal.

El uso del software 3DSlicer¹² para llevar a cabo la segmentación de las imágenes también se alinea con las normativas de la ley. Este software asegura que los datos se manipulen con precisión y que solo se acceda a la información necesaria para el desarrollo del modelo biomecánico de la próstata, minimizando así cualquier riesgo de exposición no autorizada.

La verificación de la calidad y precisión de los resultados por parte de radiólogos especializados demuestra un compromiso con la confidencialidad y exactitud de los datos. Esta validación asegura que los datos utilizados sean pertinentes y fiables, lo que contribuye a cumplir con los principios de la ley al proteger la integridad y confidencialidad de la información de los pacientes.

En resumen, el cumplimiento de la Ley de Protección de Datos Sanitarios en este estudio se refleja en la obtención de datos a través de un proceso transparente y consentido, la utilización de herramientas adecuadas para el procesamiento de datos, y la validación rigurosa de los resultados por profesionales cualificados. Esto garantiza que se respeten los derechos de privacidad y seguridad de los pacientes, así como las disposiciones legales relacionadas con la protección de datos médicos.

3.1.2 ÉTICA

En el desarrollo de este proyecto, se han abordado cuidadosamente los aspectos éticos relacionados con la obtención y el uso de datos médicos. El uso de imágenes de resonancia magnética (IRM) de pacientes para construir el modelo biomecánico de la próstata se realizó de manera ética y respetuosa, siguiendo las directrices y regulaciones establecidas por las leyes de protección de datos y la normativa de investigación médica.

Se obtuvo el consentimiento informado y firmado de cada uno de los cuatro pacientes cuyas imágenes se utilizaron en el estudio, como bien se ha expresado en el anterior apartado. Además, se implementó la anonimización de los datos para garantizar la confidencialidad y privacidad de los pacientes involucrados.

La posible discriminación por atributos protegidos, como género, raza o creencias, ha sido objeto de análisis exhaustivo. Se ha trabajado en evitar la introducción de sesgos injustos en los modelos y resultados. Al tratarse de la próstata, órgano asociado al aparato reproductor masculino, no puede haber discriminación basada en el sexo del individuo. En cuanto a los demás posibles atributos discriminantes, en principio no hay conocimientos de que la próstata tenga un comportamiento o estructura distinta según la raza, por lo que tampoco estamos siendo injustos en este aspecto.

El compromiso con la confidencialidad, la privacidad y el consentimiento informado de los pacientes ha sido central en todo el proceso. La reflexión constante sobre las

¹² <https://www.slicer.org>



Obtención de parámetros físicos mediante aprendizaje automático y algoritmos genéticos.

implicaciones éticas de cada paso del proyecto ha sido esencial para asegurar que los resultados sean confiables y socialmente responsables.

3.2 SOLUCIÓN PROPUESTA

Este proyecto tiene como objetivo principal la construcción de un algoritmo que capture con alta precisión el parámetro que rige el comportamiento biomecánico de la próstata ante diversas fuerzas aplicadas, esto es el Módulo de Young de la próstata. Este propósito se aborda en mayor detalle en la sección *1.2 Objetivos*.

El proyecto se divide en tres fases interconectadas, cada una con objetivos secundarios que contribuyen al logro final: la creación del algoritmo genético predictivo.

La primera fase se enfoca en la obtención y preparación de los datos. Aquí, se recopilarán datos que representan las geometrías específicas de las próstatas de cuatro pacientes. Utilizando el Método de Elementos Finitos (FEM), se realizarán simulaciones que abarcarán variables críticas como la intensidad de la fuerza aplicada, el punto de aplicación y las propiedades de los materiales involucrados. Este proceso proporcionará una visión integral del comportamiento inherente del órgano y arrojará luz sobre cómo los nodos se desplazan en respuesta a las fuerzas aplicadas. Los resultados extraídos de estas simulaciones FEM se integrarán en un conjunto de datos diseñado para alimentar un modelo de aprendizaje automático, más concretamente en un algoritmo genético.

La segunda fase se adentra en el campo del aprendizaje automático. Con precisión, se enfoca en la exploración, desarrollo y evaluación de un modelo predictivo que recibirá la información de una próstata y devolverá su módulo elástico. Para ello, se hará uso del flujo estándar de este tipo de algoritmo especificado en el apartado *2.3.1 Componentes, estructura y funcionamiento*.

Por último, tenemos una fase en la que se llevarán a cabo experimentos que involucrarán diversas próstatas y variaciones en los hiperparámetros para descubrir la configuración que genere los resultados más prometedores y eficientes, reduciendo así cálculos y tiempo computacional.

Esta propuesta de solución aborda la necesidad de un enfoque integral que fusiona las capacidades de la simulación y el aprendizaje automático para prever con alta precisión el comportamiento biomecánico de la próstata en respuesta a distintas fuerzas. Mediante la sinergia de estas fases, se aspira a establecer un método sólido y avanzado para la obtención de parámetros físicos cruciales en el ámbito médico.

4. PREPARACIÓN Y COMPRENSIÓN DE DATOS

En el proceso de investigar y desarrollar soluciones en el ámbito de la medicina y la tecnología, la correcta preparación y comprensión de los datos se erige como un paso fundamental. En este capítulo, se abordará exhaustivamente el proceso de recopilación y selección de datos necesarios para la creación y validación de modelos en el contexto del proyecto. La calidad y relevancia de los datos utilizados tienen un impacto directo en la precisión y confiabilidad de los resultados obtenidos. A través de un proceso riguroso, se busca garantizar que los datos reflejen de manera precisa la variabilidad y características del comportamiento biomecánico de la próstata, sentando así las bases para una investigación sólida y resultados concluyentes.

4.1 ORIGEN DE DATOS

Los datos utilizados en este proyecto se originan de las próstatas de cuatro pacientes clínicos que otorgaron su consentimiento mediante la firma correspondiente. Para obtener la información de estas próstatas, se llevaron a cabo imágenes a partir de resonancias magnéticas (IRM) después de realizar el necesario proceso de anonimización. Una vez adquiridas estas imágenes, se procedió a generar modelos tridimensionales (3D) de las próstatas mediante el empleo del software 3DSlicer, tal y como se ha detallado en la sección 3.1.1 *Análisis de la protección de datos*. Los modelos generados, conocidos como mallas, fueron almacenados en archivos de texto que contenían las coordenadas de cada uno de los nodos de dichos modelos.

Tanto la calidad como la precisión de los datos extraídos de las próstatas, incluyendo los modelos generados, fueron sometidos a validación a través de la evaluación realizada por expertos en radiología.

Una vez obtenida la representación estructural de las próstatas, se llevaron a cabo simulaciones FEM en las mallas. Los resultados de estas simulaciones proporcionan información sobre el comportamiento de la próstata ante una fuerza aplicada en un nodo específico.

Estas simulaciones se definen mediante cinco parámetros específicos que aseguran la variabilidad en cada ejecución:

- **Nodo receptor de la fuerza (f):** Dado que el número total de nodos varía en cada malla, se establece un límite en función del máximo número de nodos presentes en la geometría seleccionada. Ver *tabla 4.1*.
- **Componentes de la fuerza (f^x, f^y, f^z):** Representan el vector de fuerza que se aplica en el nodo correspondiente. En este contexto, se establece un rango mínimo de 0.0N y un máximo de 2.0N.
- **Módulo de Young del material (E_x):** Refleja la constante de elasticidad propia de la próstata. Se define un margen mínimo de $2e-3$ MPa y un margen máximo de $60e-3$ MPa.



ID Malla	Número de nodos
Malla 40	4661
Malla 42	4206
Malla 43	4007
Malla 44	4982

TABLA 4.1. NÚMERO DE NODOS EN CADA MALLA PROSTÁTICA

Estas simulaciones generan cuatro tipos de archivos distintos:

En primer lugar, se crea un archivo .log que contiene información detallada acerca de la simulación en sí, incluyendo los parámetros previamente mencionados, detalles sobre el software empleado y valores relacionados con otras variables pertinentes. En esencia, este archivo almacena la configuración y las especificaciones de la simulación generada.

Los tres archivos restantes son archivos de texto. Estos contienen información sobre el número de nodos existentes, las posiciones de los nodos y los desplazamientos resultantes al aplicar la fuerza al nodo específico indicado en el archivo .log.

4.2 GENERACIÓN DE DATOS

Los datos que se utilizan para crear el algoritmo genético es el agrupamiento de información relevante extraída a partir de las simulaciones. Para poder utilizar estos datos en la creación del algoritmo genético, es necesario crear una matriz los agrupe.

La matriz de la que estamos hablando está conformada por las siguientes variables:

- x, y, z : Coordenadas que determinan la posición del nodo cuyo desplazamiento se pretende predecir.
- f^x, f^y, f^z : Vector que denota la fuerza aplicada.
- *Material*: Propiedad inherente al material.
- n^x, n^y, n^z : Coordenadas del nodo al cual se le aplicó la fuerza.
- d^x, d^y, d^z, d^{euc} : Vectores que representan la distancia y la distancia Euclídea desde el nodo mencionado anteriormente hasta el nodo donde se aplicó la fuerza.
- $d_{min}^x, d_{min}^y, d_{min}^z, d_{min}^{euc}$: Vectores que indican la distancia y la distancia Euclídea desde el nodo descrito hasta el nodo más cercano en las tres direcciones principales.
- $d_{mean}^x, d_{mean}^y, d_{mean}^z, d_{mean}^{euc}$: Vectores que expresan la distancia y la distancia Euclídea desde el nodo descrito hasta la posición promedio de los nodos fijos en las tres direcciones.
- $d_{min,y}^x, d_{min,y}^y, d_{min,y}^z, d_{min,y}^{euc}$: Vectores que señalan la distancia y la distancia Euclídea desde el nodo descrito hasta el nodo fijo más cercano en la dirección Y.

- $d_{mean,y}^x$, $d_{mean,y}^y$, $d_{mean,y}^z$, $d_{mean,y}^{euc}$: Vectores que reflejan la distancia y la distancia Euclídea desde el nodo descrito hasta la posición media de los nodos hijos en la dirección Y.
- dim^x , dim^y , dim^z , dim^{euc} : Dimensiones de la caja de inclusión. Estas variables actúan como elementos que determinan las características geométricas
- h^x , h^y , h^z : Estas variables se generan mediante la homogeneización de las posiciones x y z, lo que permite establecer una representación más uniforme y balanceada de las coordenadas. Su incorporación en el análisis contribuye a una comprensión más precisa de la distribución espacial de los nodos y su interacción en el contexto de las simulaciones biomecánicas.

Para crear esta matriz se ha creado una función en Python a partir del código utilizado en el trabajo “Aprendizaje automático en la aceleración de cálculos biomecánicos: aplicación a la simulación del comportamiento de próstatas” ya nombrado en la sección 1.2 *Objetivos*. La función accede a la ruta donde se encuentran los ficheros generados por la simulación y crea a partir de ellos la matriz con todas las variables explicadas.

4.3 EMPLEO Y EVOLUCIÓN DE LOS DATOS

La matriz que hemos generado se utiliza para obtener una predicción de cómo se desplazan los nodos en función de la fuerza ejercida sobre uno de ellos. Esta predicción se consigue gracias al modelo de ML desarrollado en el proyecto “Aprendizaje automático en la aceleración de cálculos biomecánicos: aplicación a la simulación del comportamiento de próstatas”.

Además de la mencionada matriz y las predicciones, el algoritmo genético inicia su búsqueda del coeficiente de material mediante la creación de una población aleatoria. Esta población se compone de valores numéricos comprendidos entre $2e-3$ MPa y $60e-3$ MPa, los cuales representan las posibles magnitudes del módulo de elasticidad de la próstata. Para generar estos valores, se indica al algoritmo la cantidad de individuos a crear. Aunque también es posible especificar el tipo de distribución estadística, se ha evaluado que la distribución uniforme es la más apropiada, dado que todos los valores presentan igual probabilidad de aparición.

Finalmente, los individuos evolucionan en cada iteración del algoritmo, lo que significa que en cada ciclo se generan nuevos individuos que se espera tengan un mejor rendimiento en comparación con la población previa. En la sección 2.3 *Algoritmos genéticos* del proyecto se detallan los operadores genéticos disponibles. Con relación al funcionamiento de estos operadores dentro de nuestro algoritmo, a continuación, se presenta una explicación de cada uno de ellos:

- **Selección:** El algoritmo recibe un parámetro numérico entero que determina la cantidad de individuos de la población que pasarán a la siguiente generación. Estos individuos son elegidos en base al cálculo del error a partir de la métrica seleccionada. Si se especifica la selección de, por ejemplo, diez individuos, el algoritmo escoge los diez con mejor desempeño según la métrica.

- **Reproducción:** El segundo operador recibe una tupla (x, y) como parámetro, donde x representa los x mejores individuos e y representa los y peores individuos. Se permite la elección de individuos tanto con resultados favorables como desfavorables,



Obtención de parámetros físicos mediante aprendizaje automático y algoritmos genéticos.

ya que se ha comprobado que una reproducción entre individuos de rendimientos distintos suele agilizar la búsqueda del coeficiente.

- **Mutación:** El último operador empleado en el algoritmo recibe como parámetro el número de individuos que serán sometidos a alteración.

El funcionamiento específico de los operadores genéticos en nuestro proyecto se detalla en el siguiente capítulo, *5. Algoritmo genético*. En este se profundiza en cada uno de los métodos nombrados.

5. ALGORITMO GENÉTICO

En este apartado, se presenta en detalle el algoritmo genético desarrollado, el cual constituye la columna vertebral de este proyecto. A lo largo de esta sección, se analizarán los distintos componentes y procesos que componen este algoritmo, explicando cómo interactúan entre sí para lograr la optimización del coeficiente de material en el modelo biomecánico de la próstata. Cada etapa del algoritmo, desde la generación de la población inicial hasta la obtención de resultados convergentes, será abordada en sucesión, proporcionando una comprensión completa de su funcionamiento y su papel crucial en la obtención de parámetros físicos precisos y relevantes en el contexto médico.

El algoritmo se implementa como una clase en Python. Al utilizar esta clase, se establece una población desde el inicio, se ajustan ciertos parámetros relacionados con las operaciones genéticas y el rendimiento requerido, y se proporcionan los datos de las próstatas junto con el modelo utilizado para predecir los desplazamientos.

Por lo tanto, los parámetros aceptados por la clase **Población** son los siguientes:

- **df**: Una matriz que contiene los datos de la próstata bajo estudio.
- **Modelo**: Un modelo previamente compilado para realizar predicciones de desplazamientos.
- **n_individuos**: Número de individuos en la población inicial.
- **params_evolucion**: Una tupla de tres valores para ajustar los operadores genéticos.
 - **n_mejores**: El primer valor en la tupla define cuántos individuos se seleccionan en la fase de selección.
 - **n_cruzamiento**: Una tupla (x, y) que, como se explicó en la sección anterior, especifica tanto los mejores como los peores individuos a utilizar en la etapa de cruzamiento.
 - **n_mutacion**: El último valor indica cuántos individuos se modificarán durante la mutación.
- **params_mutacion**: Una tupla (x, dev) donde x representa la media y dev la desviación estándar de la distribución normal utilizada en la mutación.
- **params_objetivo**: Una tupla $(error, iter)$ donde $error$ es el máximo error permitido e $iter$ es el número máximo de iteraciones permitidas.

En resumen, una estructura de clase y los parámetros asociados permiten personalizar y controlar los aspectos esenciales del algoritmo genético.

5.1 INICIALIZACIÓN

La fase de **inicialización** en el algoritmo genético marca el punto de partida de la búsqueda del coeficiente físico óptimo. En esta etapa, se crean los cimientos de la población inicial que será sometida a la evolución genética. La población inicial se



compone de un conjunto de individuos, cada uno de los cuales representa una posible solución en términos del módulo de elasticidad del material prostático.

Para llevar a cabo esta inicialización, el código utiliza la clase **Población**. En esta clase se encuentran distintos métodos, entre ellos hay uno que genera una población aleatoria de individuos, cada uno representando un valor posible del módulo de elasticidad. Este proceso es fundamental para la diversidad y exploración de soluciones potenciales desde el comienzo del algoritmo. Los valores del módulo se eligen dentro de un rango predefinido, específicamente entre 0.002MPa y 0.06MPa, lo que abarca un espectro de posibles valores que podrían representar las características físicas del tejido prostático.

La etapa de **inicialización** establece la base para la búsqueda del módulo óptimo, proporcionando una variación inicial en la población de individuos. A medida que el algoritmo evoluciona, estos individuos serán sometidos a operadores genéticos para explorar y explotar soluciones potenciales de manera iterativa. La diversidad introducida en esta fase contribuye a una exploración más amplia del espacio de soluciones y puede ayudar a evitar la convergencia prematura a soluciones subóptimas.

5.2 EVALUACIÓN

El proceso inicia con la predicción de los desplazamientos de los nodos basados en el valor del módulo del material asociado a cada individuo en la población. Una vez obtenidas estas predicciones, los resultados son evaluados utilizando la métrica Mean Squared Error (MSE), que ha sido previamente explicada en la sección 2.5.1 *Error Cuadrático Medio (MSE)*. Sin embargo, además de utilizar el MSE, se añade un componente adicional a la métrica de evaluación.

Este componente adicional, conocido como el "máximo error de predicción," se centra en identificar la mayor discrepancia entre las predicciones y los valores reales. Su función principal es capturar situaciones en las que las predicciones se desvían significativamente de los valores reales, lo que podría ser especialmente crítico en aplicaciones médicas donde incluso un único error importante podría tener consecuencias significativas.

La métrica de evaluación que empleamos combina el Error Cuadrático Medio (MSE) con el "máximo error de predicción." Al hacerlo, obtenemos una medida de discrepancia más completa y sólida entre las predicciones y los valores reales. Esto se logra al considerar tanto la magnitud promedio de los errores como el peor caso de desviación. Esta métrica compuesta es esencial para identificar qué individuos en la población exhiben un rendimiento óptimo y, por lo tanto, desempeña un papel crucial en la toma de decisiones dentro del proceso de evolución genética.

Los errores de predicción junto con los individuos correspondientes se registran en listas, ordenados según su rendimiento. En este contexto, el individuo y su error asociado se encuentran en el mismo índice, pero en listas separadas.

Una vez se determina qué individuo ha tenido el mejor rendimiento, se verifica si su error es menor al permitido, el cual ha sido establecido como parámetro en la clase. Si este criterio se cumple, se considera que se ha hallado el módulo de elasticidad óptimo del material. Sin embargo, en caso de no cumplir con este requisito, se procede a comprobar si el número de iteraciones realizadas ha alcanzado el límite establecido, o si ha ocurrido una convergencia prematura. Si se da alguna de estas situaciones, indica

que no se ha conseguido un parámetro con el rendimiento deseado, lo cual puede deberse a una exigencia de rendimiento demasiado alta o a una convergencia a un valor inadecuado por parte de la población.

Finalmente, si ninguno de los casos anteriores se cumple, el proceso continúa con la evolución de la población. Esto implica la generación de nuevos individuos y resultados en un intento por alcanzar un coeficiente con el rendimiento esperado. Este ciclo de evaluación y evolución se repite hasta que se cumplan las condiciones de terminación previamente establecidas o se obtenga el coeficiente de material deseado.

5.3 EVOLUCIÓN

En esta sección se detalla la implementación de los tres operadores genéticos cruciales que modelan el proceso de selección natural y evolución. Cada uno de estos operadores contribuye a la exploración y explotación de soluciones potenciales en la población, permitiendo la formación de una nueva generación de individuos.

Selección: Este operador juega un papel fundamental en la supervivencia de los individuos con el mejor rendimiento. Se toma un número N de los mejores individuos de la población actual, según su rendimiento medido por la métrica de evaluación compuesta que hemos discutido previamente. Estos individuos seleccionados se transfieren directamente a la siguiente generación, sin cambios. En esencia, esta operación emula el proceso de selección natural, donde los individuos más adaptados tienen más probabilidades de transmitir sus características a la siguiente generación.

Cruzamiento: El operador de cruzamiento combina la información genética de los individuos seleccionados previamente. Se especifica una tupla (x, y) como parámetro para este operador, donde x representa el número de mejores individuos e y es el número de peores individuos que serán considerados. Estos individuos son emparejados uno a uno, creando diferentes combinaciones. Por ejemplo, si (x, y) es $(3, 3)$, se generan quince combinaciones posibles como se puede calcular con la siguiente fórmula de combinatoria:

$$c_m^n = \binom{m}{n} = \frac{m!}{(m-n)!n!}$$

m : total de individuos
 n : individuos por conjunto

Ecuación 5.1: Ecuación Combinatoria sin repetición.

Para cada par de individuos emparejados, se calcula la media de sus valores genéticos. Estos valores promedio se convierten en los descendientes que pasan a la siguiente fase.

Mutación: La mutación introduce variabilidad en la población al introducir cambios aleatorios en los individuos seleccionados. En este operador, se toman los M mejores individuos de la población actual. Para cada uno de estos individuos, se genera un valor aleatorio a partir de una distribución normal con una media y desviación típica especificadas como parámetros. La media suele ser un pequeño valor para asegurar una mutación gradual (por ejemplo, 0.058/20), mientras que la desviación típica controla la magnitud de la mutación (por ejemplo, 0.0001). A cada individuo seleccionado se le suma y resta el valor aleatorio generado, lo que resulta en dos nuevas versiones del



Obtención de parámetros físicos mediante aprendizaje automático y algoritmos genéticos.

individuo, es decir, dos mutaciones por cada individuo. Esta operación introduce diversidad en la población y permite explorar nuevas soluciones en el espacio de búsqueda.

Estos operadores trabajan en conjunto para dar forma a la población y permitir la evolución iterativa de los individuos. La combinación de selección, cruzamiento y mutación permite que el algoritmo busque soluciones más prometedoras y eficientes a lo largo de múltiples generaciones, impulsando la convergencia hacia soluciones óptimas o cercanas a ellas.

```
### EVOLUCIÓN GENÉTICA
def seleccion_mejores_individuos(self,ultima_poblacion):
    for n in range(0,self.n_mejores):
        self.individuos_poblacion.append(ultima_poblacion[n])

def cruzamiento_individuos(self,ultima_poblacion):
    from itertools import combinations

    lista_individuos_cruzar = []
    for n_mejor in range(0,self.n_cruzamiento[0]):
        lista_individuos_cruzar.append(ultima_poblacion[n_mejor])
    for n_peor in range(1,self.n_cruzamiento[1]+1):
        lista_individuos_cruzar.append(ultima_poblacion[-n_peor])
    for pareja in combinations(lista_individuos_cruzar,2):
        self.individuos_poblacion.append(np.mean(pareja))

def mutacion_individuos(self,ultima_poblacion):
    mutaciones_creadas = list(np.random.normal(self.media_mutacion,
                                                self.desv_mutacion,
                                                self.n_mutacion))

    for mutacion,individuo in zip(mutaciones_creadas,range(self.n_mutacion)):
        #sumamos mutación
        self.individuos_poblacion.append(ultima_poblacion[individuo]+mutacion)
        #restamos mutación
        self.individuos_poblacion.append(ultima_poblacion[individuo]-mutacion)
```

6. INVESTIGACIÓN Y RESULTADOS

En este capítulo, se exponen los resultados derivados de la investigación y experimentación llevadas a cabo para validar y evaluar el algoritmo genético propuesto en el contexto de la predicción del módulo elástico de la próstata. La implementación del algoritmo se llevó a cabo con atención a los objetivos previamente delineados, buscando así adquirir una comprensión más profunda de cómo los procesos de selección, cruzamiento y mutación pueden contribuir a una optimización efectiva. Los resultados obtenidos trascienden la mera precisión de las predicciones generadas por el algoritmo, brindando una visión esclarecedora de la eficacia de los operadores genéticos en la exploración de soluciones dentro de un espacio de parámetros amplio y complejo. En las próximas secciones, se detallan los componentes de la metodología experimental, seguidos por la presentación de los resultados y un análisis exhaustivo de los mismos.

6.1 METODOLOGÍA EXPERIMENTAL

En esta sección, se describe la metodología que se utilizó para llevar a cabo los experimentos con el algoritmo genético propuesto para la predicción del módulo elástico de la próstata. Los experimentos se diseñaron con el objetivo de evaluar el desempeño y la efectividad del algoritmo en términos de convergencia y precisión de las predicciones.

6.1.1 DISEÑO DE LOS EXPERIMENTOS

El diseño de los experimentos se basó en el uso de cuatro próstatas clínicas distintas, lo que permitió evaluar el algoritmo genético en diferentes escenarios y contextos. Para cada próstata, se crearon múltiples poblaciones iniciales, variando los parámetros clave del algoritmo. Estos parámetros incluyeron el número de individuos iniciales, el número de individuos seleccionados, el número de cruzamientos, el número de mutaciones, la media y desviación típica de la distribución para las mutaciones, el error permitido y el número máximo de iteraciones.

6.1.2 FLUJO GENERAL DE TRABAJO

El flujo general de trabajo para ejecutar los experimentos consistió en los siguientes pasos:

1. **Selección de Próstata y Poblaciones Iniciales:** Se seleccionó una próstata clínica de las cuatro disponibles para cada conjunto de experimentos. Luego, se generaron diferentes poblaciones iniciales, cada una con variaciones en los parámetros mencionados anteriormente.
2. **Ejecución del Algoritmo Genético:** Para cada población inicial, se ejecutó el algoritmo genético. Durante cada iteración del algoritmo, se llevaron a cabo operaciones de selección, cruzamiento y mutación, siguiendo los parámetros configurados para la población.
3. **Observación y Análisis:** Se observó el comportamiento de cada población a lo largo de las iteraciones. Se registraron detalles sobre la convergencia, el rendimiento de los individuos y el error alcanzado en cada iteración. Para facilitar este trabajo, está automatizada la generación de unos gráficos que muestran la evolución del



Obtención de parámetros físicos mediante aprendizaje automático y algoritmos genéticos.

parámetro material durante las distintas iteraciones y los errores asociados. Ver figura 6.1.

4. **Evaluación de Resultados:** Se evaluaron los resultados obtenidos para cada población en términos de precisión de las predicciones y la capacidad del algoritmo para converger hacia soluciones óptimas.

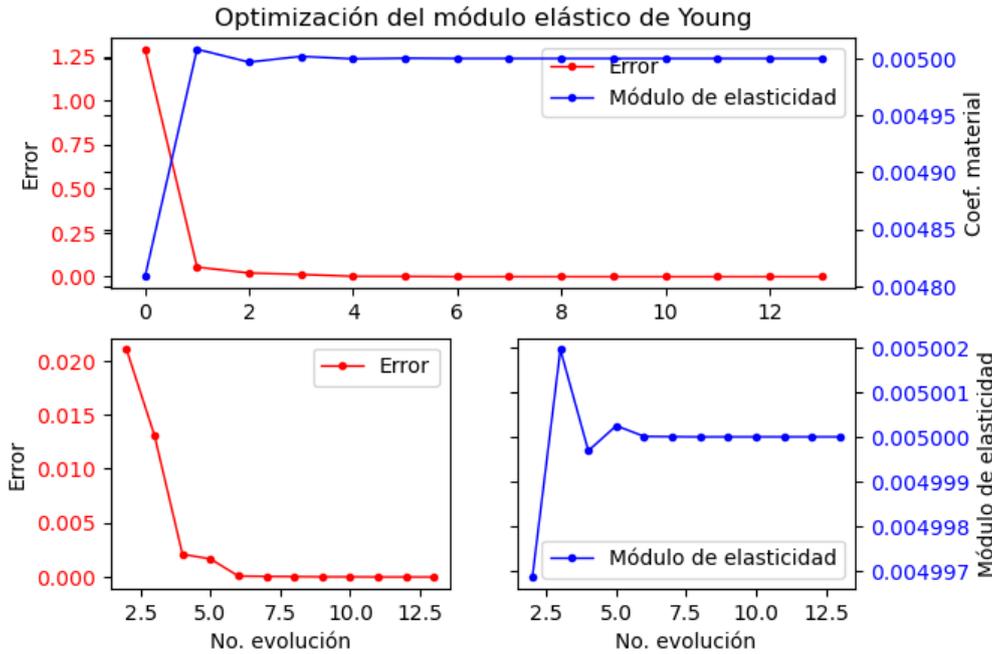


FIGURA 6.1. EVOLUCIÓN DEL MATERIAL Y EL ERROR ASOCIADO

6.1.3 DETALLES SOBRE LA CONFIGURACIÓN DE PARÁMETROS EN EL ALGORITMO GENÉTICO

La configuración de los parámetros en el algoritmo genético desempeña un papel crucial en su rendimiento y eficacia. Para alcanzar los objetivos de optimización, se llevaron a cabo diversos experimentos con el fin de encontrar la configuración óptima de los parámetros. El enfoque se centró en reducir el tiempo de procesamiento y aumentar la precisión de los resultados.

El algoritmo genético cuenta con varios tipos de parámetros, como se explicó en detalle en el capítulo 5. *Algoritmo genético*. Entre ellos, destacan tres que influyen significativamente en el desarrollo de la población y, por ende, en la búsqueda del coeficiente de elasticidad del tejido prostático.

El primer parámetro clave es $n_individuos$, que determina el tamaño de la población inicial. Esta población actúa como punto de partida para la evolución genética. La elección del valor adecuado para $n_individuos$ es fundamental, ya que puede afectar significativamente el rendimiento del algoritmo. Se llevaron a cabo experimentos variando este valor y se observó que tanto poblaciones más pequeñas como

poblaciones más grandes podían producir buenos resultados. Sin embargo, se notó que el rendimiento podía depender en gran medida de los valores iniciales generados aleatoriamente. Por ejemplo, se encontró que poblaciones iniciales de cincuenta individuos alcanzaban resultados sobresalientes, mientras que con doscientos individuos los resultados no eran óptimos, ver figura 6.2. Esto sugiere que la aleatoriedad en la inicialización puede desempeñar un papel crucial en los primeros pasos del algoritmo. Como resultado de estas pruebas, se llegó a la conclusión de que poblaciones iniciales conformadas en torno a cien individuos era suficiente. Esta cantidad proporcionaba un equilibrio entre la diversidad necesaria para explorar diversas soluciones y la mitigación del impacto de la aleatoriedad en los valores iniciales. Además, este rango de tamaños de población no influyó significativamente en el tiempo de ejecución del algoritmo, ya que generaba una cantidad manejable de datos.

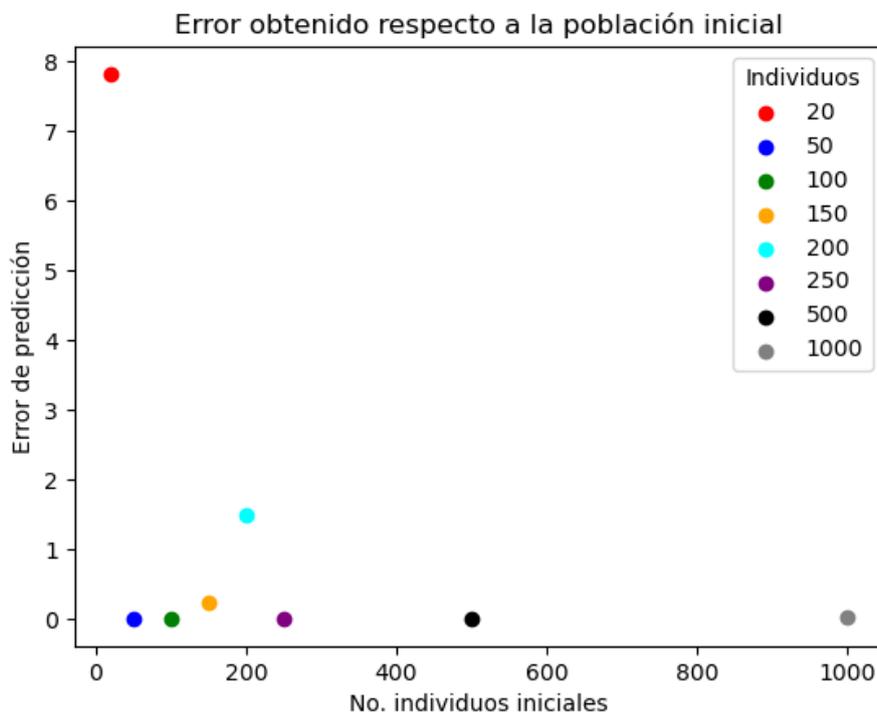


FIGURA 6.2. ERROR DE PREDICCIÓN SEGÚN LA POBLACIÓN INICIAL

La optimización de los operadores genéticos fue otro aspecto fundamental. Para ello, nos inspiramos en estudios en los que se presentan una metodología exhaustiva para la optimización de parámetros en este tipo de algoritmos ([8]). El parámetro *params_evolucion* engloba tres opciones que afectan al proceso de selección, cruzamiento y mutación de individuos. En el caso de la selección, se determinó que seleccionar el 25% de los mejores individuos de la población era una estrategia eficaz. Esto permitía mantener una diversidad adecuada mientras se avanzaba hacia soluciones más prometedoras. En cuanto al cruzamiento, se descubrió que seleccionar entre cinco y ocho individuos tanto de alto rendimiento como de bajo rendimiento para la reproducción generaba una variabilidad valiosa en las nuevas poblaciones. Esta variabilidad no solo exploraba soluciones cercanas a los óptimos locales sino también soluciones intermedias. Por último, se implementó un mecanismo para introducir variabilidad en torno a las mejores soluciones al elegir el 10% de los individuos con mejores métricas y aplicar dos mutaciones a cada uno de ellos.



Obtención de parámetros físicos mediante aprendizaje automático y algoritmos genéticos.

El parámetro *params_mutacion* fue clave para ajustar la magnitud de las mutaciones en los individuos. La media se estableció en 0.0193, una tercera parte del rango de valores posibles (0.058), para garantizar mutaciones suaves en individuos que ya estaban cerca de soluciones óptimas. La desviación típica se configuró para permitir cambios sutiles en los valores, asegurando que las mutaciones no fueran drásticas.

En resumen, la configuración de parámetros en el algoritmo genético se basó en experimentación rigurosa y consideraciones teóricas. Se buscó alcanzar un equilibrio entre la optimización de resultados y la eficiencia temporal. Estas configuraciones permitieron explorar un amplio rango de soluciones, mejorar la convergencia y garantizar la precisión en la búsqueda del coeficiente de elasticidad del tejido prostático.

7. CONCLUSIONES

En este trabajo de investigación, se ha abordado el desafío de predecir el módulo de elasticidad en tejido prostático mediante una combinación innovadora de algoritmos genéticos y modelos de aprendizaje automático. A lo largo de este proceso, hemos avanzado en la comprensión de cómo estas dos disciplinas pueden colaborar para abordar un problema complejo en el ámbito de la ingeniería biomédica.

La relación entre los avances en cálculo biomecánico y la optimización computacional se ha explorado a fondo. Si bien el método de los elementos finitos (FEM) es reconocido por su precisión en la simulación de comportamiento biomecánico, su coste computacional y la complejidad de obtener los parámetros adecuados han limitado su aplicabilidad. En este contexto, la utilización de algoritmos genéticos como herramienta para buscar soluciones óptimas en un espacio de parámetros amplio ha demostrado ser una aproximación efectiva.

La implementación y evaluación del algoritmo genético propuesto han arrojado resultados prometedores. La capacidad del algoritmo para converger hacia soluciones que se ajustan a los valores reales del módulo de elasticidad en el tejido prostático subraya su eficacia en la optimización de parámetros. Además, la incorporación de modelos de aprendizaje automático para acelerar los cálculos FEM ha demostrado ser una estrategia exitosa para reducir significativamente el tiempo de procesamiento, lo que es esencial en entornos clínicos y de simulación.

El enfoque iterativo del algoritmo genético, junto con la evaluación de distintos conjuntos de parámetros, ha permitido explorar un amplio rango de soluciones y analizar su rendimiento. A través de estos experimentos, se han identificado combinaciones de parámetros que conducen a resultados óptimos y se han obtenido valiosas conclusiones sobre cómo ajustar estos valores para maximizar la convergencia y precisión del algoritmo.

Es importante destacar que este enfoque ha demostrado ser significativamente más rápido que los métodos tradicionales de FEM utilizados en trabajos anteriores ([10]). Esto es especialmente relevante en entornos clínicos donde se requieren resultados más rápidos para la toma de decisiones médicas.

Sin embargo, es importante reconocer que este trabajo abre la puerta a múltiples direcciones futuras. El análisis de interpretación y visualización de los resultados, la incorporación de técnicas más avanzadas de aprendizaje automático y la aplicación en contextos clínicos reales son áreas que pueden profundizarse en investigaciones posteriores. Este trabajo sienta las bases para un mayor desarrollo y aplicación práctica en la ingeniería biomédica y la atención médica.

En última instancia, esta investigación destaca la sinergia entre la optimización evolutiva y el aprendizaje automático en el campo de la ingeniería biomédica. La capacidad de abordar problemas de optimización en entornos complejos y la posibilidad de acelerar cálculos computacionales intensivos son logros significativos que pueden impactar positivamente en el diagnóstico, tratamiento y diseño de dispositivos médicos. Este trabajo representa un paso hacia adelante en la dirección de la innovación y la mejora de la atención médica a través de la tecnología y la ciencia de datos.

7.1 RELACIÓN DEL TRABAJO CON LOS ESTUDIOS CURSADOS



Obtención de parámetros físicos mediante aprendizaje automático y algoritmos genéticos.

En este apartado, se establecerá la relación del trabajo realizado con los estudios cursados a lo largo del programa académico. El proyecto de investigación fusiona conocimientos adquiridos en diversas materias, abordando de manera integral aspectos de diferentes disciplinas. A continuación, se detalla cómo los contenidos y habilidades de los cursos han influido y respaldado la realización de este trabajo:

- **Análisis Numérico y Métodos Computacionales:** Los conceptos adquiridos en este curso en relación con la resolución numérica de ecuaciones diferenciales han sido fundamentales para comprender los fundamentos matemáticos del algoritmo genético desarrollado. La habilidad para manejar métodos de discretización y aproximación ha sido aplicada en la implementación de la población inicial y en la evaluación iterativa de las soluciones.
- **Programación y Algorítmica:** Me ha proporcionado una sólida base en la creación y estructuración de algoritmos, así como en la implementación eficiente de soluciones mediante lenguajes de programación como Python. La habilidad para escribir código limpio, modular y eficiente ha sido esencial en la creación de los algoritmos genéticos y la interacción con los modelos de aprendizaje automático que forman parte integral de este trabajo.
- **Aprendizaje Automático y Minería de Datos:** La comprensión de los conceptos fundamentales del aprendizaje automático y tratamiento de datos ha sido clave para la utilización de la red neuronal utilizada para la obtención de datos. Además, los conocimientos sobre minería de datos han sido aplicados a la hora de estructurar y utilizar los distintos datos, tanto los empleados para el algoritmo como la información obtenida durante las evoluciones poblacionales.
- **Optimización y Métodos Heurísticos:** El entendimiento de los métodos de optimización y su aplicación en la resolución de problemas complejos ha sido esencial para el diseño y ajuste de los operadores genéticos. Los principios de búsqueda de soluciones óptimas en espacios de parámetros amplios han sido aplicados en la estrategia evolutiva del algoritmo genético.

En resumen, este trabajo de investigación se nutre de los conocimientos adquiridos en diversas áreas del programa académico. La convergencia de conceptos de análisis numérico, programación, aprendizaje automático y optimización ha permitido abordar de manera integral el desafío de acelerar los cálculos FEM y obtener los parámetros biomecánicos necesarios para la simulación. La intersección de estos campos ha enriquecido la solidez teórica y práctica del proyecto, demostrando la aplicación y profundización de los contenidos estudiados durante el programa.

7.2 TRABAJO FUTUROS

Una dirección prometedora para futuras investigaciones es la optimización y adaptación de los parámetros en los algoritmos genéticos. Aunque se han realizado experimentos en este trabajo para examinar el impacto de los parámetros en el rendimiento del algoritmo, una exploración más exhaustiva podría proporcionar una comprensión más profunda de cómo ajustar estos valores para maximizar la convergencia y precisión.

Otro aspecto interesante sería aplicar este enfoque a otros tipos de tejidos biológicos o problemas de ingeniería biomédica, ampliando así el alcance de su aplicabilidad.

Además, se podría explorar la integración de datos multimodales, como imágenes médicas, para mejorar aún más la precisión de las predicciones.

En términos de la metodología, sería valioso investigar cómo abordar la interpretación y visualización de los resultados obtenidos. Explorar técnicas que ayuden a comprender las relaciones entre los parámetros de los individuos y sus correspondientes predicciones podría proporcionar una visión más profunda de los factores que influyen en el comportamiento biomecánico.

En conclusión, este trabajo de investigación abre múltiples caminos para la exploración futura en el campo de la predicción de propiedades biomecánicas. Las mejoras en los algoritmos, la incorporación de técnicas de aprendizaje automático más avanzadas y la aplicación de metodologías enfocadas a la interpretación y visualización son solo algunas de las direcciones emocionantes que podrían seguirse para ampliar y enriquecer aún más este campo de estudio.



BIBLIOGRAFÍA

- [1] García Pastor, A. (2023). Aprendizaje automático en la aceleración de cálculos biomecánicos: aplicación a la simulación del comportamiento de próstatas. Universitat Politècnica de València. <http://hdl.handle.net/10251/194404>
- [2] García Reyes, Alonso. (2016). Sobre el mecanismo de volteo en taludes y laderas, enfoque numérico aplicado con elemento finito. 10.13140/RG.2.2.22333.87521.
- [3] R. K. Singh, V. K. Panchal and B. K. Singh, "A review on Genetic Algorithm and Its Applications," *2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT)*, Bangalore, India, 2018, pp. 376-380, doi: 10.1109/ICGCIoT.2018.8753030.
- [4] Alouani AT, Elfouly T. Traumatic Brain Injury (TBI) Detection: Past, Present, and Future. *Biomedicines*. 2022; 10(10):2472. <https://doi.org/10.3390/biomedicines10102472>
- [5] Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos y por el que se deroga la Directiva 95/46/CE (Reglamento general de protección de datos). *Diario Oficial de la Unión Europea, L119/1*, de 27 de abril de 2016. <https://www.boe.es/doue/2016/119/L00001-00088.pdf>
- [6] Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales. *Boletín Oficial del Estado, 294*, de 06 de diciembre de 2018. <https://www.boe.es/eli/es/lo/2018/12/05/3/con>
- [7] Ley 41/2002, de 14 de noviembre, básica reguladora de la autonomía del paciente y de derechos y obligaciones en materia de información y documentación clínica. *Boletín Oficial del Estado, 274*, del 15 de noviembre de 2002. <https://www.boe.es/eli/es/l/2002/11/14/41/con>
- [8] McCall, J. (2005). Genetic algorithms for modelling and optimisation. *Journal of Computational and Applied Mathematics*, 184(1), 205-222. <https://doi.org/10.1016/j.cam.2004.07.034>
- [9] Wan Ishak WH, Siraj F. (2000). "Artificial intelligence in medicine." (s. f.). <http://groups.csail.mit.edu/medg/people/psz/ftp/AIM82/>
- [10] Martínez Martínez, F. (2014). Determining the Biomechanical Behavior of the Liver Using Medical Image Analysis and Evolutionary Computation [Tesis doctoral no publicada]. Universitat Politècnica de València. <https://doi.org/10.4995/Thesis/10251/39337>
- [11] Botchkarev, A. (2018). Performance Metrics (Error Measures) in Machine Learning Regression, Forecasting and Prognostics: Properties and Typology. *arXiv (Cornell University)*. <https://doi.org/10.28945/4184>

- [12] Eiben, A. E., & Smit, S. (2011). Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and evolutionary computation*, 1(1), 19-31. <https://doi.org/10.1016/j.swevo.2011.02.001>
- [13] Chebbi, O., & Siala, J.C. (2015). Effective parameter tuning for genetic algorithm to solve a real-world transportation problem. *2015 20th International Conference on Methods and Models in Automation and Robotics (MMAR)*, 370-375.
- [14] Iglesias, A., Gálvez, A. (2023). Genetic Algorithm Applications for Challenging Real-World Problems: Some Recent Advances and Future Trends. In: Dey, N. (eds) *Applied Genetic Algorithm and Its Variants*. Springer Tracts in Nature-Inspired Computing. Springer, Singapore. https://doi.org/10.1007/978-981-99-3428-7_2
- [15] Kumar, Y., Koul, A., Singla, R. *et al.* Artificial intelligence in disease diagnosis: a systematic literature review, synthesizing framework and future research agenda. *J Ambient Intell Human Comput* 14, 8459–8486 (2023). <https://doi.org/10.1007/s12652-021-03612-z>



Obtención de parámetros físicos mediante aprendizaje automático y algoritmos genéticos.

APÉNDICE A – ALGORITMO GENÉTICO

class Poblacion:

```
def __init__(self, df, modelo, n_individuos, params_evolucion, params_mutacion, params_objetivo):
    import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import tensorflow as tf
    import keras
    import tensorflow_addons as tfa
    import time

    self.df_referencia = df
    self.modelo_predictivo = modelo
    self.desplazamientos_referencia = self.modelo_predictivo.predict(df.astype('float64'), verbose = 0)
    self.n_evolucion = 0 # Contador de número de iteraciones

    # Arrays donde almacenar los datos obtenidos para las estimaciones de coeficiente físico (material)
    # -- los asociados al menor error de cada "generación" -- y sus errores en cada evolución de la
    # población.
    self.mejores_materiales = []
    self.mejores_materiales_errores = []

    # Arrays para la población ACTUAL (materiales)
    self.individuos_poblacion = []
    self.errores_poblacion = []
    self.predicciones_poblacion = []

    # Parámetros poblacionales y genéticos
    self.tamano = n_individuos
    self.n_mejores = params_evolucion[0]
    self.n_cruzamiento = params_evolucion[1]
    self.n_mutacion = params_evolucion[2]

    # Parámetros media y desviación típica de la mutación
    self.media_mutacion = params_mutacion[0]
    self.desv_mutacion = params_mutacion[1]

    # Parámetros del resultado permitido
    self.error_permitido = params_objetivo[0]
    self.iter_permitida = params_objetivo[1]
    self.estancamiento_permitido = params_objetivo[2]

    # Arrays donde almacenar tiempos de ejecución de los operadores genéticos y del proceso global
    self.coste_temporal_seleccion = []
    self.coste_temporal_cruzamiento = []
    self.coste_temporal_mutacion = []
    self.coste_temporal_poblacion = []

    def obtener_mejor_individuo(self):
        mejor_material = self.mejores_materiales[-1]
        mejor_material_error = self.mejores_materiales_errores[-1]
        return (mejor_material, mejor_material_error)

    def crear_poblacion(self):
        self.individuos_poblacion = list(np.random.uniform(0.002, 0.06, self.tamano))

    def predecir_desplazamientos(self):
```



Obtención de parámetros físicos mediante aprendizaje automático y algoritmos genéticos.

```
for material in self.individuos_poblacion:
    datos_poblacion = self.df_referencia.copy()
    datos_poblacion.loc[:, 'material'] = material
    datos_poblacion = datos_poblacion.astype(np.float64)
    prediccion = self.modelo_predictivo.predict(datos_poblacion, verbose = 0)
    self.predicciones_poblacion.append(prediccion)

def evaluar_predicciones(self):
    for desplazamientos_prediccion in self.predicciones_poblacion:
        error_prediccion = np.square(np.subtract(self.desplazamientos_referencia,
desplazamientos_prediccion)).mean() + \
            np.max(np.abs(np.subtract(self.desplazamientos_referencia,
desplazamientos_prediccion)))
        selferrores_poblacion.append(error_prediccion)
        # No ordenamos predicciones_poblacion porque se calcularán en cada iteración
        self.errores_poblacion, self.individuos_poblacion = map(list,
zip(*sorted(zip(self.errores_poblacion, self.individuos_poblacion))))
        self.mejores_materiales.append(self.individuos_poblacion[0])
        self.mejores_materiales_errores.append(self.errores_poblacion[0])

### EVOLUCIÓN GENÉTICA
def seleccion_mejores_individuos(self, ultima_poblacion):
    for n in range(0, self.n_mejores):
        self.individuos_poblacion.append(ultima_poblacion[n])

def cruzamiento_individuos(self, ultima_poblacion):
    from itertools import combinations

    lista_individuos_cruzar = []
    for n_mejor in range(0, self.n_cruzamiento[0]):
        lista_individuos_cruzar.append(ultima_poblacion[n_mejor])
    for n_peor in range(1, self.n_cruzamiento[1]+1):
        lista_individuos_cruzar.append(ultima_poblacion[-n_peor])
    for pareja in combinations(lista_individuos_cruzar, 2):
        self.individuos_poblacion.append(np.mean(pareja))

def mutacion_individuos(self, ultima_poblacion):
    mutaciones_creadas = list(np.random.normal(self.media_mutacion, self.desv_mutacion,
self.n_mutacion))
    for mutacion, individuo in zip(mutaciones_creadas, range(self.n_mutacion)):
        self.individuos_poblacion.append(ultima_poblacion[individuo]+mutacion) #sumamos mutación
        self.individuos_poblacion.append(ultima_poblacion[individuo]-mutacion) #restamos mutación

## FUNCIÓN PARA CREAR SIGUIENTE POBLACIÓN
def desarrollo_genetico(self):
    individuos_anteriores = self.individuos_poblacion.copy() # lista auxiliar de la última población
    self.individuos_poblacion = []
    self.errores_poblacion = []
    self.predicciones_poblacion = []
    t_operador1 = time.time()
    self.seleccion_mejores_individuos(individuos_anteriores)
    t_operador2 = time.time()
    self.cruzamiento_individuos(individuos_anteriores)
    t_operador3 = time.time()
    self.mutacion_individuos(individuos_anteriores)
```



```

t_4 = time.time()
self.coste_temporal_seleccion.append(t_operador2-t_operador1)
self.coste_temporal_cruzamiento.append(t_operador3-t_operador2)
self.coste_temporal_mutacion.append(t_4-t_operador3)

def check_resultado_optimo(self):
    print(f'Material actual: {self.obtener_mejor_individuo()[0]}, Error actual:
{self.obtener_mejor_individuo()[1]}, Error permitido: {self.error_permitido}')
    return self.obtener_mejor_individuo()[1]<self.error_permitido

def evolucion_poblacion(self):
    self.crear_poblacion()
    while self.n_evolucion<=self.iter_permitida:
        t_ini = time.time()
        print(f'EVOLUCIÓN NÚMERO: {self.n_evolucion+1}, Población: {len(self.individuos_poblacion)}
personas')
        self.predecir_desplazamientos()
        self.evaluar_predicciones()
        if self.check_resultado_optimo():
            self.coste_temporal_poblacion.append(time.time()-t_ini)
            return self.obtener_mejor_individuo()
        elif (len(self.mejores_materiales)>self.estancamiento_permitido and
len(set(self.mejores_materiales[-self.estancamiento_permitido:]))==1):
            print(f'Finalización temprana. El coeficiente se ha estancado durante
{self.estancamiento_permitido} iteraciones')
            self.coste_temporal_poblacion.append(time.time()-t_ini)
            return self.obtener_mejor_individuo()
        self.desarrollo_genetico()
        t_fin = time.time()
        self.coste_temporal_poblacion.append(t_fin-t_ini)
        self.n_evolucion += 1
    return self.obtener_mejor_individuo()

def analisis_material(self):
    def two_scales(ax1, time, data1, data2, c1, c2):

        ax2 = ax1.twinx()

        j0 = ax1.plot(time, data1, c1, markersize=3,linewidth=1, label = 'Error')
        ax1.set_ylabel('Error')

        j1 = ax2.plot(time, data2, c2, markersize=3,linewidth=1, label = 'Módulo de elasticidad')
        ax2.set_ylabel('Coef. material')
        ax2.ticklabel_format(useOffset=False, style='plain')
        ax2.yaxis.set_ticks_position('both')
        lns = j0+j1
        labs = [l.get_label() for l in lns]
        ax1.legend(lns, labs, loc=0)
        return ax1, ax2

    def zoom_scales(ax1,ax2, time, data1, data2, c1, c2):

        ax1.plot(time, data1, c1, markersize=3,linewidth=1, label = 'Error')
        ax1.set_ylabel('Error')

```



Obtención de parámetros físicos mediante aprendizaje automático y algoritmos genéticos.

```
ax2.plot(time, data2, c2, markersize=3,linewidth=1, label = 'Módulo de elasticidad')
ax2.set_ylabel('Módulo de elasticidad')
ax2.yaxis.set_label_position("right")
ax2.ticklabel_format(useOffset=False, style='plain')
ax2.yaxis.tick_right()
ax2.yaxis.set_ticks_position("both")

return ax1, ax2

# Change color of each axis
def color_y_axis(ax, color):
    """Color your axes."""
    for t in ax.get_yticklabels():
        t.set_color(color)
    return None

# Create axes
fig = plt.figure()
axis1 = fig.add_subplot(2, 1, 1)
ax3 = fig.add_subplot(2, 2, 3)
ax4 = fig.add_subplot(2, 2, 4)

t = list(range(len(self.mejores_materiales_errores)))
ax1, ax2 = two_scales(axis1, t, self.mejores_materiales_errores, self.mejores_materiales, 'ro-', 'bo-')

list_diff = np.diff(self.mejores_materiales_errores)
lim = next((i for i,diff in enumerate(list_diff) if ((abs(diff) < 0.01) and i>1)),2)

ax3, ax4 = zoom_scales(ax3,ax4, t[lim:], self.mejores_materiales_errores[lim:],
self.mejores_materiales[lim:], 'ro-', 'bo-')

color_y_axis(ax1, 'r')
color_y_axis(ax2, 'b')
color_y_axis(ax3, 'r')
color_y_axis(ax4, 'b')

ax3.set_xlabel('No. evolución')
ax4.set_xlabel('No. evolución')

ax3.legend()
ax4.legend()

plt.title('Optimización del módulo elástico de Young')
plt.show()

def analisis_temporal(self):
    tiempo_pobl = np.cumsum(self.coste_temporal_poblacion)
    tiempo_selec = np.cumsum(self.coste_temporal_seleccion)
    tiempo_cruz = np.cumsum(self.coste_temporal_cruzamiento)
    tiempo_mut = np.cumsum(self.coste_temporal_mutacion)
    return (tiempo_pobl,tiempo_selec,tiempo_cruz,tiempo_mut)

def analisis_total(self):
    tiempos = self.analisis_temporal()[0]
```

```
num_iteracion = self.n_evolucion
num_individuos = len(self.individuos_poblacion)
mejor_resultado = self.obtener_mejor_individuo()
return {'tiempos':tiempos, 'num_iter':num_iteracion,'num_individuos_final':
num_individuos,'mejor_resultado': mejor_resultado}
```



Obtención de parámetros físicos mediante aprendizaje automático y algoritmos genéticos.

APÉNDICE B – OBJETIVOS DEL DESARROLLO SOSTENIBLE (ODS)



ANEXO OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Proced e
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.	X			
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.			X	
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.		X		
ODS 9. Industria, innovación e infraestructuras.	X			
ODS 10. Reducción de las desigualdades.		X		
ODS 11. Ciudades y comunidades sostenibles.				X
ODS 12. Producción y consumo responsables.				X
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.		X		

Obtención de parámetros físicos mediante aprendizaje automático y algoritmos genéticos.

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

A continuación, analizamos cómo este proyecto se relaciona con algunos de los Objetivos de Desarrollo Sostenible (ODS):

- **Salud y bienestar (ODS 3):** Este proyecto se centra en el ámbito de la medicina al utilizar tecnologías avanzadas como el aprendizaje automático para comprender y predecir el comportamiento de la próstata en respuesta a fuerzas. Al mejorar la precisión en la simulación de tejidos y órganos, se contribuye al avance de la medicina, lo que a su vez puede mejorar la atención médica, el diagnóstico y el tratamiento, promoviendo así la salud y el bienestar.
- **Igualdad de género (ODS 5):** Si bien el proyecto se enfoca en la próstata, un órgano asociado al sistema reproductor masculino, sus avances pueden tener un impacto en la equidad de género en la medicina. Al desarrollar métodos más precisos y avanzados para simular y comprender el comportamiento de órganos, se sienta un precedente para futuras investigaciones en el campo de la salud femenina, promoviendo la igualdad de género en el ámbito médico.
- **Trabajo decente y crecimiento económico (ODS 8):** La implementación de tecnologías como el aprendizaje automático y los algoritmos genéticos en el campo médico no solo contribuye a la investigación, sino que también puede impulsar la creación de empleo en la investigación médica y la industria de la salud. Además, los avances en la comprensión de los tejidos y órganos pueden llevar a tratamientos más efectivos, lo que a su vez puede impulsar el crecimiento económico en el sector de la salud.
- **Industria, innovación e infraestructuras (ODS 9):** Este proyecto se alinea directamente con el ODS 9 al utilizar tecnologías de vanguardia para innovar en la simulación y predicción de comportamientos biomecánicos. La integración de algoritmos genéticos en el ámbito médico representa un avance en la infraestructura tecnológica y científica, contribuyendo a la innovación en la industria médica.
- **Reducción de las desigualdades (ODS 10):** Al mejorar la precisión en la simulación de órganos y tejidos, este proyecto puede tener un impacto positivo en la atención médica para diversas poblaciones. La reducción de las desigualdades en el acceso a tratamientos más precisos y efectivos puede resultar de los avances logrados a través de esta investigación.
- **Alianzas para lograr objetivos (ODS 17):** La integración de conocimientos y habilidades de diversos campos demuestra la importancia de las alianzas multidisciplinarias en la consecución de objetivos globales, como el avance en la atención médica y la mejora de la calidad de vida.

En resumen, este proyecto no solo se adentra en la comprensión biomecánica, sino que también tiene el potencial de impactar positivamente en múltiples áreas de desarrollo sostenible, desde la salud y la igualdad de género hasta la innovación y las alianzas colaborativas.