

**FEDERAL UNIVERSITY OF MINAS GERAIS
DEPARTMENT OF ELECTRICAL ENGINEERING**

FELIPE ARAGÃO NOGUEIRA DE FREITAS

**MIXING DEEP NEURAL NETWORK MODEL AND GENETIC ALGORITHM FOR
PLASMONIC BIOSENSORS DESIGN**

BELO HORIZONTE

2022

FELIPE ARAGÃO NOGUEIRA DE FREITAS

MIXING DEEP NEURAL NETWORK MODEL AND GENETIC ALGORITHM FOR
PLASMONIC BIOSENSORS DESIGN

Undergraduate Thesis submitted to the Electrical Engineering course of the Federal University of Minas Gerais, as a partial requirement for obtaining the Bachelor Degree in Electrical Engineering.

Advisor: Prof. Dr. Jhonattan Córdoba
Ramírez

BELO HORIZONTE

2022

FELIPE ARAGÃO NOGUEIRA DE FREITAS

MIXING DEEP NEURAL NETWORK MODEL AND GENETIC ALGORITHM FOR
PLASMONIC BIOSENSORS DESIGN

Undergraduate Thesis submitted to the Electrical Engineering course of the Federal University of Minas Gerais, as a partial requirement for obtaining the Bachelor Degree in Electrical Engineering.

Approved on:

EXAMINATION BOARD

Prof. Dr. Jhonattan Córdoba Ramírez (Advisor)
Federal University of Minas Gerais (UFMG)

Prof. Dr. Omar Paranaíba Vilela Neto
Federal University of Minas Gerais (UFMG)

Prof. Dr. Frederico Gualberto Ferreira Coelho
Federal University of Minas Gerais (UFMG)

To my family and friends for their unwavering support and partnership over these years.

ACKNOWLEDGEMENTS

My heartfelt gratitude goes to my parents, Ana Paula and Ricardo, my sister, Júlia, and my beloved Vera. For holding my hand and unconditionally supporting me over the last five years. To my colleagues and friends who stood by me during the most difficult times of my Electrical Engineering graduation.

A special thank you to my supervisor, Jhonattan Córdoba Ramírez, for your dedication, support, and availability. Your work as a professional and educator motivates your students to continue researching with zeal and enthusiasm. A thank you to OptMA Lab, Light Propagation and Photonics Group, and NanoComp Lab for your support and structure that were critical to enable the studies to go forward.

Finally, I want to thank UFMG and Mendes Pimentel Foundation (FUMP) for sponsoring me throughout my graduation as a Level 1 student in the assistance programs; the financial support was critical in allowing me to give the important dedication that this course requires.

“Knowledge is power? No. Knowledge on its own is nothing, but the application of useful knowledge, now that is powerful.”

(Rob Liano)

ABSTRACT

The biosensors market is a high-growth and relevant market worth USD\$25,5 billion in 2021, with a positive forecast of USD\$36,7 billion in 2026. Real-time diagnosis and home care devices are currently the most widely available biosensors, requiring high efficiency and sensitivity as well as low cost and power consumption. In this scenario, optical biosensors, particularly those based on Localized Surface Plasmon Resonance (LSPR), promote miniaturization as they are based on nanoparticle technology, allow label-free sample analysis, and are experimentally low cost and efficient. An LSPR-based biosensor detects the presence of specific proteins in biological samples through the interaction between a light source and the free electrons on the device surface. To project this type of sensor, which is highly shape sensitive, numerical methods such as Finite-Difference Time-Domain (FDTD) are typically used through software simulation. The issue is that finding and optimizing sensor geometry requires a significant investment in computational resources and time. Furthermore, the high sensitive geometries are frequently non-intuitive, necessitating additional numerical tests. As a result, this project proposed and implemented a class-based Python framework to automate the simulation and outcome analysis process. This framework also includes a Machine Learning intelligent model that combines a Genetic Algorithm and a Deep Neural Network to optimize and recommend non-intuitive device geometries that promote high biosensor sensitivity and efficiency. Therefore, this document describes the steps involved in developing and validating the framework as well as the intelligent model, and also how it performed in comparison to the traditional random search and parametric sweep methods. Finally, it shows how this apparatus was used to improve in 55% of an LSPR-based biosensor experimental performance while significantly reducing time and resource demand.

Keywords: LSPR. Biosensor. Plasmonic. DNN. Genetic Algorithm. Surrogate Model.

RESUMO

O mercado de biosensores vem crescendo de maneira expressiva estando avaliado em USD\$25,5 bilhões em 2021 e com previsão de USD\$36,7 bilhões em 2026. Dispositivos para diagnóstico em tempo real e cuidados domiciliares são os mais utilizados atualmente, demandando alta eficiência, precisão, eficiência e baixo consumo.

Nesse cenário, sensores óticos, em especial aqueles baseados em Ressonância Plasmônica de Superfície Localizada (LSPR), promovem a miniaturização dos dispositivos por serem baseados em tecnologia de nanopartículas, permitem análises sem rotulagem, e são experimentalmente mais baratos e eficientes. Um sensor baseado em LSPR detecta a presença de proteínas em amostras biológicas por meio da interação de uma fonte de luz com os elétrons livres na superfície do dispositivo.

O projeto deste tipo de sensor, altamente sensíveis à sua geometria, usualmente utiliza métodos numéricos como Diferenças-Finitas no Domínio do Tempo (FDTD) por meio de simulações computacionais. O grande problema é o fato de que encontrar e otimizar geometrias para esses sensores demanda grandes valores de recurso computacional e tempo. Além disso, geometrias que promovem alta eficiência são, muitas vezes, não intuitivas, demandando ainda mais testes via simulação. Desse modo, neste projeto foi proposto e implementado um *framework* em Python baseado em classes, para automatizar o processo de design e análise de resultado das simulações numéricas.

Este *framework* ainda inclui um modelo de Aprendizado de Máquina que combina um Algoritmo Genético e uma Rede Neural de Aprendizado Profundo que juntos promovem a otimização e recomendação de geometrias não intuitivas gerando aumento relevante na sensibilidade e eficiência de biosensores.

Portanto, este documento descreve os passos para desenvolver e validar tanto o *framework* quanto o modelo inteligente assim como apresenta análise comparativa entre o modelo e métodos de busca tradicionais como busca aleatória e força bruta.

Por fim, é apresentado como este arcabouço foi aplicado melhorando em 55% a performance experimental de um dispositivo baseado em LSPR enquanto diminuindo a demanda por recurso computacional e tempo gasto em simulação.

Palavras-chave: LSPR. Biossensor. Plasmonica. DNN. Algoritmo Genético.

LIST OF FIGURES

Figure 1 – Graph: Design Feasibility vs Spectral Complexity	23
Figure 2 – SPR Description	26
Figure 3 – LSPR Description	27
Figure 4 – LSPR-Based Biosensor	27
Figure 5 – Genetic Algorithm Cycle	31
Figure 6 – Roulette Wheel Selection.	32
Figure 7 – K-way Tournament Selection.	32
Figure 8 – Single Neuron Neural Network.	34
Figure 9 – Multi-Layer Neural Network.	35
Figure 10 – NN Most Used Activation Functions.	36
Figure 11 – DNN Training and Application Pipeline.	38
Figure 12 – DNA Variation Measure.	39
Figure 13 – LSPR-based Detection Pipeline.	40
Figure 14 – LSPR-based Detection Pipeline.	41
Figure 15 – Subwavelength Lattice.	42
Figure 16 – SPR-based Optimized Geometry.	43
Figure 17 – SPR Different Geometries.	44
Figure 18 – Makiel’s Bidirectional DNNs.	44
Figure 19 – Project Pipeline	47
Figure 20 – Design flow diagram.	47
Figure 21 – Sensor Base Geometry and Simulation Apparatus.	51
Figure 22 – Base Geometry Periodic Representation.	51
Figure 23 – Framework Class Diagram.	53
Figure 24 – Complete Design Flow Diagram.	54
Figure 25 – GA Application Procedure Pipeline.	59
Figure 26 – Results Pipeline.	61
Figure 27 – Base Geometry and Absorption Curve.	62
Figure 28 – Parametric Sweep Heat Map.	63
Figure 29 – Parametric Sweep Best Geometry.	64
Figure 30 – Parametric Sweep 2 nd Best Geometry.	64
Figure 31 – Random Search Summary.	65

Figure 32 – Best Single-Peak Random Geometry. 65

Figure 33 – GA Methods Outcome. 68

Figure 34 – Final Genetic Algorithm Evolution. 69

Figure 35 – Optimized Geometries Absorption Curves. 70

Figure 36 – Search Space Exploration. 70

Figure 37 – Fabrication Process. 71

Figure 38 – Fabricated Device. 72

LIST OF TABLES

Table 1 – Experimental Flow Summary	57
Table 2 – DNN Grid Search Summary	66
Table 3 – DNN Training Size Sensitivity	67
Table 4 – Intelligent Model Final Architecture	68
Table 5 – Final Set Summary	69

LIST OF ABBREVIATIONS AND ACRONYMS

API	Application Programming Interface
DNN	Deep Neural Network
FDTD	Finite-Difference Time-Domain
FPS	Fitness Proportionate Selection
GA	Genetic Algorithm
IM	Intelligent Model
LSPR	Localized Surface Plasmon Resonance
NN	Neural Networks

LIST OF SYMBOLS

$\Delta\lambda$	Deslocamento no pico da curva de ressonância
-----------------	--

CONTENTS

1	INTRODUCTION	21
1.1	Overview	21
1.2	Motivation	22
1.3	Objectives	23
2	THEORETICAL BACKGROUND	25
2.1	LSPR-Based Biosensors	25
2.2	Finite-Difference Time-Domain Method	28
2.3	Genetic Algorithm Flow	30
2.4	Surrogate Model and Neural Networks	33
3	LITERATURE REVIEW	39
3.1	State of the Art on LSPR-Based Biosensor	39
3.2	AI-Powered Biosensor Design	42
4	METHODOLOGY	47
4.1	Sensor Design Flow	47
4.2	Base Geometry Development	50
4.3	Automation Framework Development	52
4.4	Intelligent Model Development	54
4.4.1	Genetic Algorithm	54
4.4.2	NN Surrogate Model	56
4.5	Intelligent Model Application	57
5	RESULTS AND DISCUSSION	61
5.1	Manually Obtained Base Geometry	61
5.2	Automatically Obtained Geometries	62
5.3	Smartly Obtained Optimized Geometries	66
5.4	Device Fabrication	71
6	CONCLUSION	73
6.1	Accomplishments	73
6.2	Future Work	73
	REFERENCES	75

1 INTRODUCTION

Combining multiple areas of knowledge is a powerful tool for scientific evolution. In this research, we propose combining computational tools and intelligent methods to model and analyze optical physical behavior that can indicate the presence of a specific analyte in biological samples.

As a result, this chapter provides an overview of the current market and forecast for biosensors, explains the actual design methods for these sensors, and provides the necessary theoretical background for understanding the new development flow we proposed, as well as all of the motivations for this research.

1.1 Overview

An analyte is an element, like a protein, that can be present in a biological sample as blood or spit. The existence of specific analyte in a sample can lead to many diseases diagnosis like COVID-19, Dengue, Zika and many others. In addition, biosensors are devices that can iterate with those samples in order to identify analyte's presence in multiple ways.

The biosensors market was valued at USD\$25,5 billion in 2021, with a positive forecast of USD\$36,7 billion in 2026. This is twice as representative as it was six years ago, and it is growing rapidly due to the demands brought on by the COVID pandemic, as well as relevant advances in plasmonic nanosensors. (MARKET, 2021)

Real-time diagnosis and home care devices are currently the most widely available biosensors on the market. They expect high efficiency and sensitivity, as well as low cost and power consumption. As a result, optical biosensors offer numerous advantages in this scenario.

Localized Surface Plasmon Resonance (LSPR) sensors, in particular, promote miniaturization because they are based on nanoparticle technology, in addition to enabling label-free sample analysis. For these reasons, many researchers are focusing on LSPR technology, including this work, which aims to improve the modeling and design flow for LSPR-based biosensors. (LOPEZ *et al.*, 2017)

There are several methods for modeling and estimating the optical behavior of LSPR sensors. The analytical method is based on Maxwell's equations, which is impractical to use in a project as the geometries become more complex. Another method is to use computationally implemented numerical methods, such as the Finite Elements Method (FEM) and Finite-Difference

Time-Domain Method (FDTD).

Although the numerical approach provides interesting precision and reality proximity, using numerical simulations in a design process requires an expressive amount of time and computational resources. Aside from that, it necessitates a designer who is well-versed in optical fundamentals and expected behavior.

Intelligent algorithms can be introduced in a variety of ways to improve the design flow and mitigate time and resource consuming issues. Some of the techniques with a high potential for improving the design flow are evolutionary algorithms and neural network models.(MALKIEL *et al.*, 2018)

As a result, the challenges of automating the LSPR design process and incorporating computational intelligent algorithms into the project flow will be addressed in the following chapters of this document.

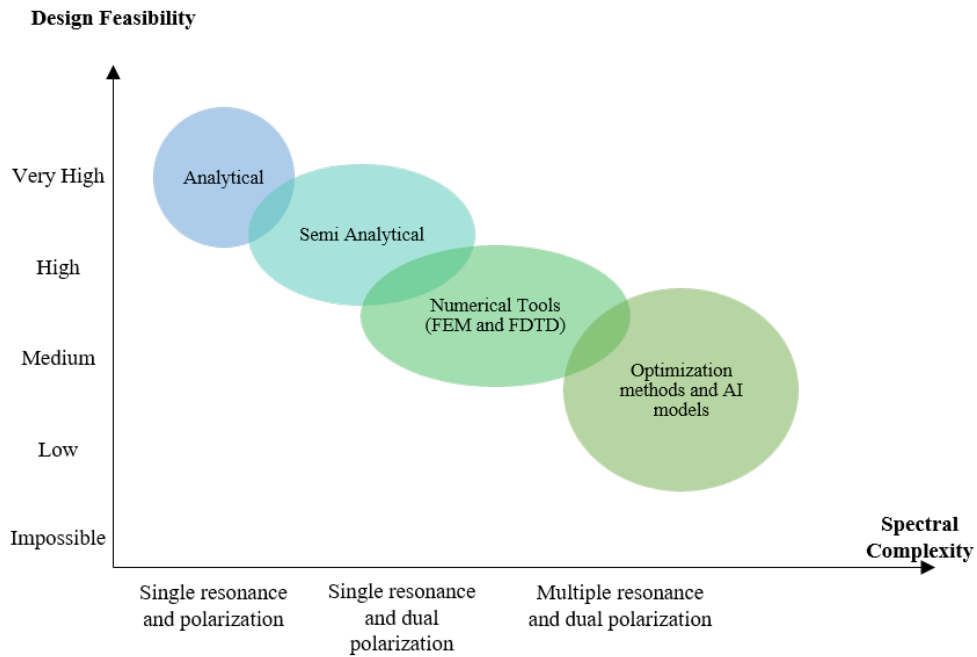
1.2 Motivation

As previously stated, the biosensors market, particularly for optical devices, is emerging and attracting the attention of many researchers worldwide. There is still much to discover and many opportunities for advancement and innovation.

To outperform actual plasmonic nanosensors in this scenario, the design process must be optimized, and intelligent algorithms emerge as a solution. The relationship between device geometry feasibility (Y) and spectral complexity (X) is depicted in Figure 1. Where spectral complexity represents the sensor's optical behavior and circles represent the most appropriate techniques to be introduced in the project.

The main motivations for this work are the challenges of dealing with complex designs and the need for an automated and intelligent flow in order to propose better and more precise LSPR biosensors.

Figure 1 – Graph: Design Feasibility vs Spectral Complexity



Adapted from (MALKIEL *et al.*, 2018)

1.3 Objectives

The primary goal of this work is to create a computational and intelligent method that combines a Genetic Algorithm with a Neural Network model to enable the development of an optimized set of LSPR nanosensors. Other specific objectives include:

- To create a Python framework that is integrated with the numerical simulator ©Ansys Lumerical in order to automate simulation execution and data analysis in the sensor design flow.
- To implement a Genetic Algorithm (GA) and include it in the Python framework as an optimizer for the proposed set of sensors.
- To design and build a Neural Network (NN) model capable of learning numerical simulation behavior and serving as a surrogate model for the GA.
- To use the entire framework and optimization flow to create a collection of LSPR sensor designs.
- To validate the set of LSPR sensors proposed in a high precision numerical simulation.

2 THEORETICAL BACKGROUND

This chapter summarizes the major concepts discussed and applied during this research. It starts by highlighting the nuances of LSPR-based biosensors used to detect the presence of specific analytes in biological samples, as well as the numerical methods used to model these devices. Finally, we describe the computational intelligent methods used to improve the design flow of LSPR-based biosensors, which include a Genetic Algorithm as a parameter search space method and a Deep Neural Network as a surrogate model that replaces the majority of numerical simulations during the evolutionary flow.

2.1 LSPR-Based Biosensors

Localized Surface Plasmon Resonance (LSPR)-based biosensors are highly sensitive, low-cost, and power-efficient devices that provide numerous benefits such as promoting device miniaturization and enabling label-free sample analysis.

LSPR is a result of the interaction of light and metal nanoparticles (NPs), but before we can understand the nuances of an LSPR-Based sensor, we must first examine its foundation in the Surface Plasmon Resonance (SPR).

Plasmon Resonance is a phenomenon that occurs when a free electron in a metal surface interacts with incident light and begins to oscillate at the frequency of the incident electromagnetic field. Surface Plasmon Resonance is the result of electron oscillations in a metal-dielectric interface.

In the SPR field, incident light passes through a thin metal film, typically made of $\approx 200nm$ gold or plate, with a specific angle of incidence that determines how much of the electrical magnetic wave is absorbed and transferred to electron oscillations. Called resonance angle is the angle that causes electrons to resonate.

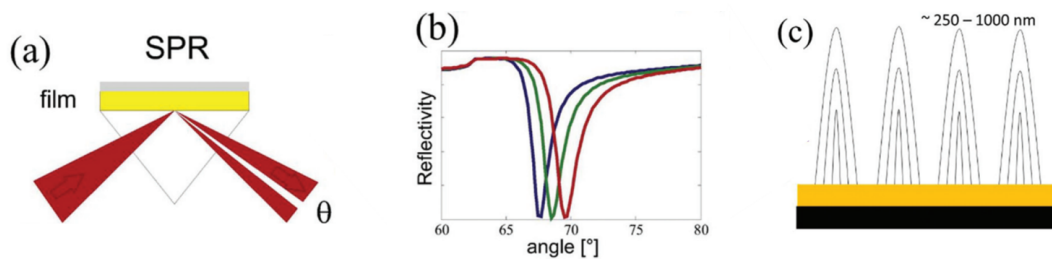
When there is a modification in the surface of a metallic film that changes its refractive index, the level of absorption changes as well, and the resonance peak occurs at a new angle value. This modification could be represented by the insertion of a bio molecule on the film's surface, the presence of which is quantified by the shift between the old and new resonance angle values. This is the basic idea behind biosensing via SPR devices.

Figure 2 depicts the behavior of the SPR device. (a) and (b) depict the relationship between different incident angles, which represent different positions for the peak of electron

resonance. While (c) shows the evanescent field produced by the plasmon phenomenon. The decay length of this evanescent field defines the level of device sensitivity.

SPR is known as high sensitive technology due to its longer decay length, which is interesting but can become a problem when it begins to be affected by noise in the vicinity of the field, which can change the refractive index in the material but is not related to the objective sample in analysis. This is called bulk effect.

Figure 2 – SPR Description



Source: Adapted from (LIU *et al.*, 2020)

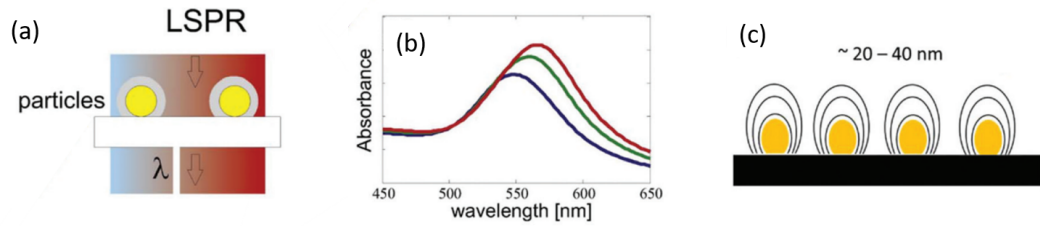
In addition to being sensitive to the bulk effect, the entire process of using an SPR device is costly and necessitates specialized complex hardware and machinery. Because it is based on the angle of incidence, it is also highly susceptible to noise and vibration interference.

As a result, a new technique has emerged to address some of these issues. This is the Localized Surface Plasmon Resonance (LSPR), which is made of metal NPs, rather than a thin metal film, that are effectively separated from each other and deposited in a substrate.

Using metal NPs reduces the intensity of the evanescent field and compress it around each NPs. Making the resonance to be localized on those NPs. It has some advantages such as improving the sensitivity for molecular binding which is the phenomena of iteration and coupling between two molecules via its surface electrons. This enables the analysis for small biological molecules.

Figure 3 illustrates the behavior of an LSPR device. The incident angle is no longer an important parameter, and different devices will have different absorption peaks for different light wavelengths. This type of device operates in the visible light spectrum, from 400nm to 700nm. This shift in the absorption peak can occur in devices due to their shape, material, or the presence of an analyte on their surface.

Figure 3 – LSPR Description

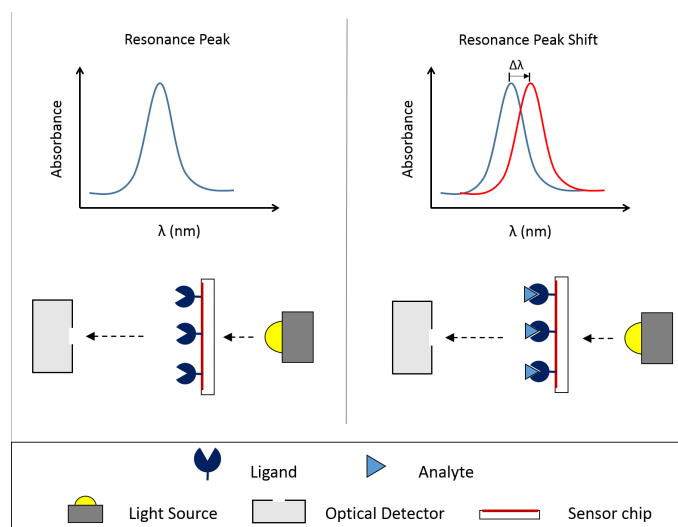


Source: Adapted from (LIU *et al.*, 2020)

When two or more NPs are deposited close to each other, they become susceptible to a phenomenon known as plasmonic coupling. This phenomenon causes each particle's single resonance curve to behave as a combination in a hybrid resonance. This phenomenon also increases the evanescent field intensity between the NPs, increasing sensitivity in that space and allowing for high precision detection of small variations on the device refractive index.

Having defined the major concepts underlying LSPR sensors, it is now possible to combine them into the definition of an analyte detection apparatus, as illustrated by figure 4. The sensor chip on both sides of the image is a representation of a golden NP-based LSPR device. The ligand is a structure that allows the analyte to bind with the NPs. Finally, a light source stimulates the device, promoting the resonance curves.

Figure 4 – LSPR-Based Biosensor



Source: (LIFESCIENCES, 2016)

On the left, it is seeing the device without an analyte and its pure resonance curve. The wavelength absorption peak on the right side has shifted red. This shift is quantified and reports the analyte's presence. Equation 2.1 depicts the quantification of the peak shift.

$$\Delta\lambda \approx m(n_{adsorbate} - n_{medium})(1 - e^{\frac{2d}{l_d}}) \quad (2.1)$$

Where m is a sensitivity factor in nm by RIU, $n_{adsorbate}$ and n_{medium} are the respectively refractive indices of adsorbents and of the original medium, d is the effective thickness of the adsorbed layer, and l_d is the evanescent field decay length.

A more precise sensor is one that returns higher $\Delta\lambda$ values for smaller differences between $n_{adsorbate}$ and n_{medium} . So, in order to improve an LSPR-based biosensor, it is obvious that m and d , which are device shape and material dependent, must be optimized, which is accomplished by converging to an ideal sensor geometry.

When these parameters are optimized, the absorption peak will be more concentrated in a single wavelength value with a high intensity and narrow peak, promoting notable peak shifts for smaller $n_{adsorbate}$ and n_{medium} deltas. Therefore, seeking an optimized device entails seeking an intense and narrow absorption curve.

A variety of techniques can be used to model the optical behavior and design the sensor geometry; the analytical way to model electromagnetic behavior is through Maxwell's equations. However, the analytical approach is not appropriate for device geometries with high efficiency. As a result, Finite-difference time-domain (FDTD) is a widely used numerical method for calculating three-dimensional optical devices, which will be briefly described in the following section.

2.2 Finite-Difference Time-Domain Method

It is beyond the scope of this document to go into detail about the numerical methods used in electromagnetic modeling using computational software, but it is critical to understand the foundation of the method we used in our numerical experiments. As a result, this section summarizes the Finite-Difference Time-Domain (FDTD) method and its key parameters.

FDTD is a popular method for solving Maxwell's equations and, as a result, modeling optical nanometric devices. The method works by discretizing space and time, treating the device

geometry as a mesh of finite-differences but solving it in the time domain. The mathematical apparatus for FDTD is summarized below; A full demonstration is provided by (CELA, 2014).

Maxwell's Equation, 2.2 and 2.3, in its derivative form, is used to model the Circular Electric field generated by a time oscillating Magnetic Field and the Circular Magnetic field generated by a time oscillating Electric Field.

$$\frac{\partial \mathbf{E}}{\partial t} = \frac{1}{\epsilon_0} \nabla \times \mathbf{H} \quad (2.2)$$

$$\frac{\partial \mathbf{H}}{\partial t} = -\frac{1}{\mu_0} \nabla \times \mathbf{E} \quad (2.3)$$

FDTD derives a discrete version of these equations based on time and space steps. These are parameters that are related to one another; the space steps are also known as mesh steps, and they represent the precision of the finite-difference between device geometry points. Looking at equations 2.4 and 2.5, we can see a discrete version of 2.2 and 2.3 for the E x and H y components.

$$\frac{E_x^{n+1/2}(k) - E_x^{n-1/2}(k)}{\Delta t} = -\frac{1}{\epsilon_0} \frac{H_y^n(k+1/2) - H_y^n(k-1/2)}{\Delta z} \quad (2.4)$$

$$\frac{H_y^{n+1}(k+1/2) - H_y^n(k+1/2)}{\Delta t} = -\frac{1}{\mu_0} \frac{E_x^{n+1/2}(k+1) - E_x^{n+1/2}(k)}{\Delta z}. \quad (2.5)$$

The derivative with respect to time applied to a point k in space is shown on the left side of this equation. It is computed as the central difference of two time step points. The right side depicts the central difference in the space based on the Δz value.

Finally, equations 2.6 and 2.7, details a software implementable version of 2.4 and 2.5 with a normalization in E amplitude so that it is close to H amplitude.

$$E_x^{n+1/2}(k) = E_x^{n-1/2}(k) + \frac{1}{\sqrt{\mu_0 \epsilon_0}} \frac{\Delta t}{\Delta z} (H_y^n(k-1/2) - H_y^n(k+1/2)) \quad (2.6)$$

$$H_y^{n+1}(k+1/2) = H_y^n(k+1/2) + \frac{1}{\sqrt{\mu_0 \epsilon_0}} \frac{\Delta t}{\Delta z} (E_x^{n+1/2}(k) - E_x^{n+1/2}(k+1)) \quad (2.7)$$

These equations demonstrate that the calculus of E or H for the next time point $n + \frac{1}{2}$ is determined by its value at the previous time point, the relationship between time and mesh precision, and the central difference for E and H in space. However, because all of these values can be obtained in a discrete manner, the equations can be implemented computationally. It also requires an initialization electric or magnetic field, known as the excitation source. Then, in a cyclical fashion, one field derives from the other.

Two major drawbacks of using this numerical approach in a software-based environment are that more memory space is required for each space or time point of the geometry and simulation. Complex geometries and excitation sources necessitate a higher level of mesh step density and, as a result, more simulation time points.

All of these issues point to the need to reduce the number of numerical simulations used in the design process. The issue is that it is impossible to eliminate the need for testing multiple geometries, which would necessitate hundreds of numerical trials in a pure simulation scenario, in order to select an effective device.

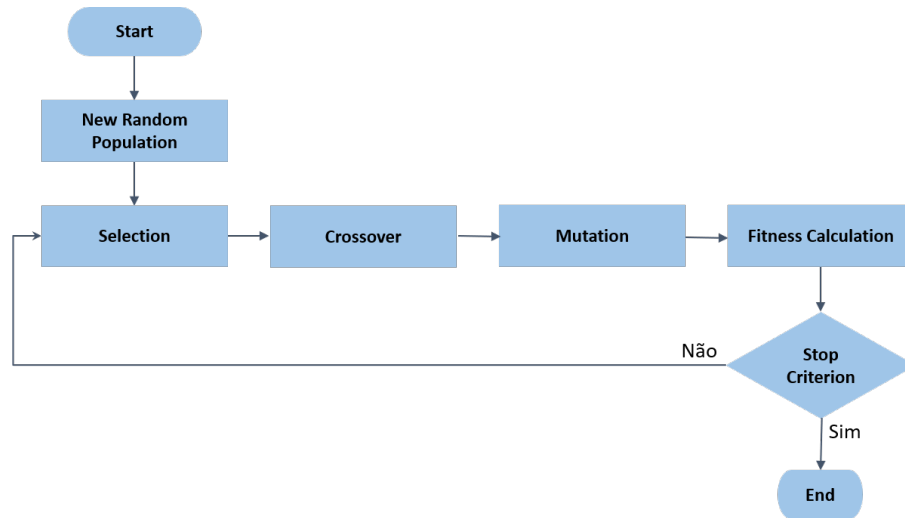
One goal of our research is to mitigate these issues by first automating the process of creating and testing new geometries devices and avoiding multiple simulation trials by using a surrogate model that learns numerical behavior from some previous simulations and quickly predicts the behavior for new suggested geometries.

Our model combines two machine learning techniques: genetic algorithms and neural networks, the main concepts of which are discussed in the following two sections.

2.3 Genetic Algorithm Flow

A Genetic Algorithm (GA) is a computer model of natural evolution in which individuals with a specific set of characteristics (genes) continue to proliferate in the population and improve these features while the remainder of the population goes extinct. This is a classic search space strategy that is employed in complex optimization situations where a satisfactory solution must be found in a short amount of time. Figure 5 gives an overview of a generic GA flow.

Figure 5 – Genetic Algorithm Cycle



Source: Author

One advantage of GAs is that they deliver not only an optimized individual but also an optimized population. An **Individual** is the subject of the optimization that is the structure or process that the GA aims to improve. An individual is unique and it is described by its genes. In our research an individual is a representation of one possible biosensor geometry.

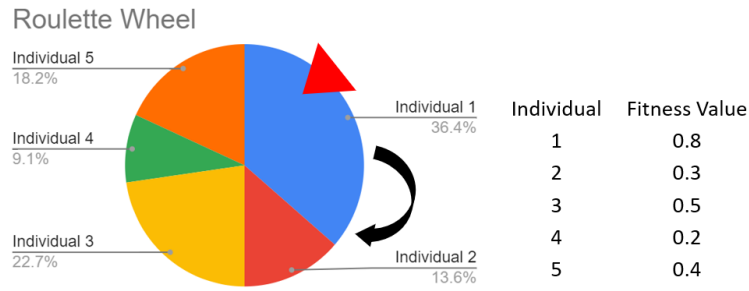
A group of individuals composes a **Population**, which needs to have an initialization method that is often a random choice for individuals genes. This population changes with time, creating a new **Generation**, which computationally is seen as an iteration of the GA method in which a new collection of individuals is formed by combining the earlier ones.

The GA stages of selection, cross-over, and mutation are repeated to create a new generation. Each of these stages can be carried out in a variety of ways; different methods for carrying out these stages will be more appropriate for various types of problems. That is why it is important to comprehend both the methods and the problems. The following are the most commonly used methods for each step:

Selection: Beginning with the initial population, selection is the stage in which two individuals are chosen to be united as parents. There are several methods for selecting parents, the most common of which are:

- **Fitness Proportionate Selection (FPS):** It is a method that increases the chances of the best fitness parents being chosen. The roulette wheel is a common FPS method. In this technique, each individual is given a weight based on its fitness, and the method then makes a random choice influenced by the individual's probability list.

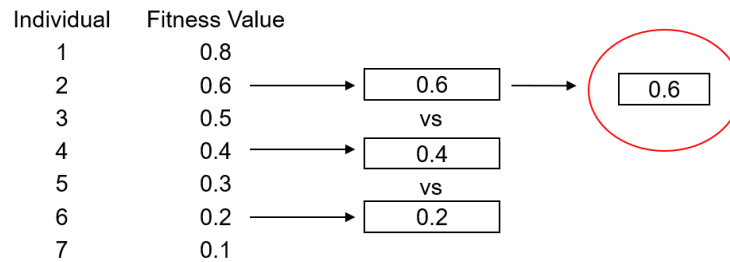
Figure 6 – Roulette Wheel Selection.



Source: Author

- **Ranking Selection:** It is similar to a roulette wheel, but instead of weighing an individual's fitness, it takes into account the individual's rank in a population. The main benefit of ranking selection occurs when the entire population begins to converge to extremely close fitness values, at which point all individuals have a similar choice probability with FPS.
- **K-way Tournament Selection:** In this method, k individuals from the population are chosen at random, and the one with the best fitness is chosen to be the parent. The second parent is chosen in the same manner.

Figure 7 – K-way Tournament Selection.



Source: Author

Crossover: This is the process in which the genes from the chosen parents are combined. The crossover procedures are known as crossover operators, and the most common ones are:

- **Linear Crossover:** A linear crossover is a linear combination of genes from both parents. It is defined by equations 2.8 and 2.9.

$$C1 = \alpha P1_genes + (1 - \alpha) P2_genes \quad (2.8)$$

$$C2 = (1 - \alpha)P1_genes + \alpha P2_genes \quad (2.9)$$

Where α is a random chosen number between 0 and 1 that represents the proportion of each parent to each child. And P1 or P2 genes represent the array of distinct values for each geometrical aspect.

- **K Point Crossover:** K point crossover is produced by switching values between parents at k separate positions in a gene list.
- **Uniform Crossover:** It is possible to swap or not swap the values of each gene. This decision can be made at random or with a bias.

Mutation: Its primary purpose is to implant new features in a generation. It is only used on a few new offspring. A mutation in binary gene representation can be defined as a change in bit state in a specific gene. A random reset for a parameter can be used for continuous numerical representation.

After generating a new individual, its representative value must be defined. It is the **fitness calculation** process. The fitness is frequently represented by a mathematical function, and optimizing an individual entails either maximizing or minimizing this function. Fitness computation is a recurring phase for each new offspring and can be quite expensive if the function is difficult to evaluate or incorporates other methods, such as numerical simulation.

This is the case in our research when a numerical simulation is required for fitness calculation. A surrogate model, which depicts the function behavior with high accuracy but is substantially less expensive to compute, is one solution.

The next section defines a surrogate model. It also covers the fundamental ideas of Neural Networks, which are used in this study's surrogate model suggestion.

2.4 Surrogate Model and Neural Networks

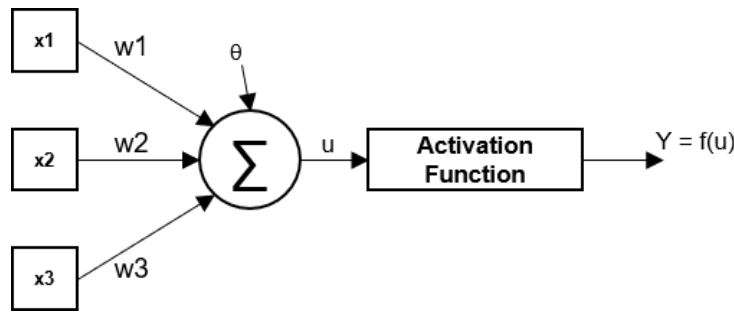
A Surrogate Model is a statistical tool that allows the modeling of a complex system in a simpler or less expensive model. It must co-relate model inputs to outputs with a certain degree of accuracy. A Surrogate Model could be represented by a Neural Network that precisely predicts an output based on the model's input with a certain precision grade and statistical validation.

Neural Networks Neural Networks (NN) are widely used machine learning techniques. In the context of supervised learning, where the primary goal is to learn patterns based

on previous experiences and provide classifications or predictions for new ones, this is a computational representation of the human brain in which each neuron can be updated as the network learning process progresses.

A learning process represents the network training, and a training data set is a collection of experiences. This data set is made up of a sufficient number of instances that can be considered samples. Each sample has distinct characteristics that will be used as inputs for the NN model. Figure 8 depicts a visual representation of a single-layer NN.

Figure 8 – Single Neuron Neural Network.



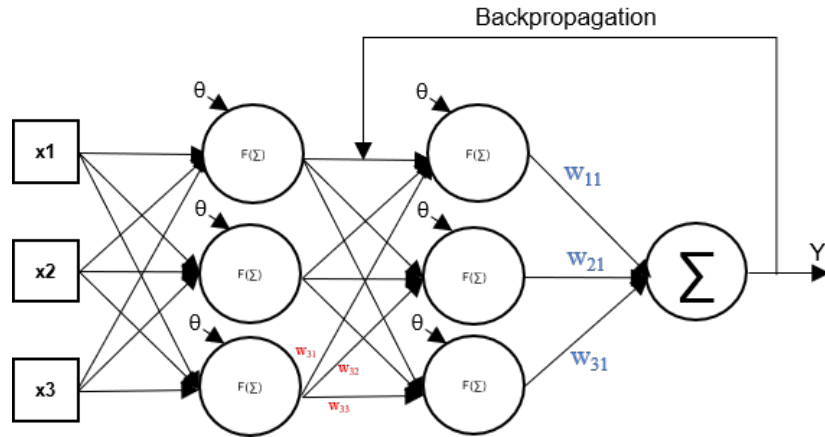
Source: Author

Where x_1 , x_2 , and x_3 are the sample X 's attributes. These inputs will be combined in a linear sum weighed by w_1 , w_2 , w_3 and θ yielding a value u that will be used as an input to an activation function. Finally, the output of the activation function is Y , which can be a classification of x or a prediction of x value based on its attributes. The predicted and expected Y values are compared, and the error is used to update the model weights. Equation 2.10 shows how u is obtained.

$$u = \sum_{i=1}^n w_i x_i + \theta \quad (2.10)$$

A NN can have multiple internal layers known as hidden layers. Figure 9 depicts a simple example of a NN with three inputs, one output, and two hidden layers. The combination of multiple layers in a NN allows for the creation of Deep Neural Networks Deep Neural Network (DNN), which are far more intelligent systems capable of learning and understanding features for the most complex modeled phenomena.

Figure 9 – Multi-Layer Neural Network.



Source: Author

As a result, by combining multiple layers, the activation function, and weight update methods, a powerful intelligent model capable of understanding nontrivial behaviors such as those encountered in the optical modeling problem can be created. To better understand how these hidden layers interact, we following describe the most commonly used activation function and the Backpropagation method for updating weights.

The literature shows a wide range of possible activation functions, with the following being the most commonly used:

Linear Function: this is a simple linear relation between the input and output, the identity function itself could be used, as seen in equation 2.11. This can yield interesting results for linear classification problems, but as the problem becomes more complex, nonlinear methods become more effective.

$$f(x) = x \quad (2.11)$$

Sigmoid Function: The sigmoid function has a S shape, and its derivative is another continuous function that is important for weights updating methods. Because the output of a Sigmoid function is a value between 0 and 1, it is commonly used for probabilistic problems.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.12)$$

Hyperbolic Tangent Function: Similar to the Sigmoid function, but with an output

range of -1 to 1. This 0 centered behavior allows for a more accurate classification of highly negative, neutral, and highly positive outputs.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.13)$$

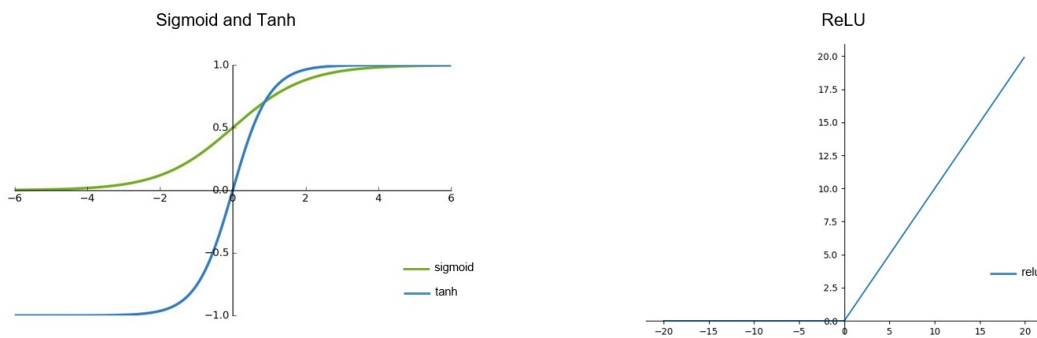
The Sigmoid and Tanh functions have some limitations; they provide good sensitivity near their centers, but it begins to saturate near their limit values, making them less effective. They are also restricted to the vanishing gradient problem (HOCHREITER, 1998), a well-known phenomenon that reduces the effective learning process via weights updates.

ReLU Function: The Rectified Linear Unit (ReLU) function is the most commonly used activation function in deep learning processes today. This occurs because ReLU overcomes some of the Sigmoid and Tanh limitations, resulting in a mix of linear and nonlinear benefits. When the value of the linear summation is greater than zero, ReLU returns the identity value, otherwise it returns 0. It enables a less expensive training flow by activating and computing neurons that provide a positive value for the activation function. It avoids the saturation problem while remaining a derivable function.

$$f(x) = \max\{0, x\} \quad (2.14)$$

Figure 10 presents a visual representation for the nonlinear functions discussed.

Figure 10 – NN Most Used Activation Functions.



Source: Author

As these functions are derivable, the Backpropagation (BP) method can be used to update weights during the learning process. BP is a popular method because it propagates errors

backwards into the model's hidden layers by calculating the gradient decedent of the model loss function with respect to the model's weight.

A loss function is a model accuracy measure that is typically a relationship between the predicted Y value and the expected correct value. A commonly used loss function is the sum of the squared errors seen in equation 2.15.

$$SSE = \frac{1}{2} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (2.15)$$

From this loss function, we can describe the BP algorithm. A complete demonstration of BP can be seen on (BRAGA *et al.*, 2007), here we synthesized the two main equations that describes BP behavior. Equation 2.16 describes how the output layer weights are updated:

$$\mathbf{W}_{ij} = \mathbf{W}_{ij} + \eta e_j f'(u_j) h_i(\mathbf{X}_t, \mathbf{Z}_i) \quad (2.16)$$

Where W_{ij} is the weight from the connection of the neuron i on the last hidden layer connected to the output j , η is a constant known as learning rate that can be experimentally defined, e_j is the error for the output j , $f'(u_j)$ is the activation function derivative applied to the output and $h_i(\mathbf{X}_t, \mathbf{Z}_i)$ is the output of the neuron i . Figure 9 shows W_{ij} in blue.

Equation 2.17, for other side, presents the update relation for the neurons in the hidden layers:

$$\mathbf{W}_{lj} = \mathbf{W}_{lj} + \eta e_j h'(u_{kj}) x_l \quad (2.17)$$

This equations is similar to the previous one, except that each neuron j in the hidden layer is treated as an output for the previous hidden layer. Therefore, W_{ij} is the weight of the connection between a neuron l to a neuron j in the next hidden layer. As result, instead of calculating the output activation function derivative, it is calculated the neuron j activation function derivative. Also, x_l is the output of a neuron l in the previous hidden layer. If it is the first hidden layer, x_l represents the value for attribute l from the input sample. Figure 9 shows W_{lj} in red.

Before presenting the entire learning pipeline, one final important concept to mention is data preparation. When representing different aspects of a sample, input attributes may have a different order of magnitude. As a result, it may bias the learning process if some attributes are

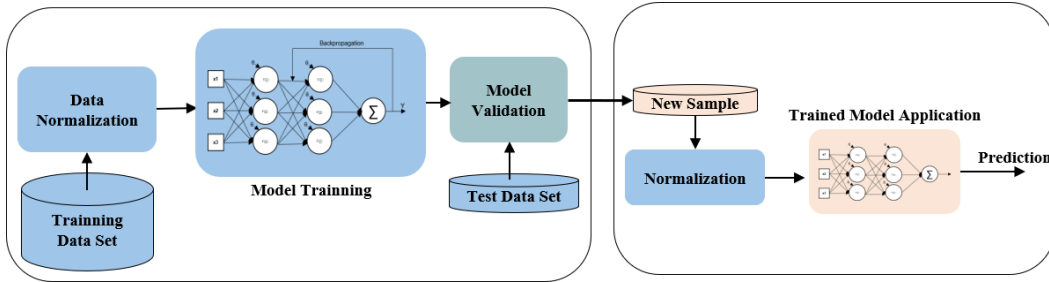
more important than others. To avoid this, normalization methods are applied to give attributes the same weight during the neurons training. Equation 2.18 defines normalization.

$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2.18)$$

Where X_{new} is the new value of the sample attribute, X_{max} and X_{min} are respectively the maximum and minimum value of that attribute between all samples in the data set.

Finally, we can combine all of the concepts and methods discussed in a learning pipeline, also known as a machine learning pipeline, as shown in figure 11. The first step is to generate training data and pre-process it. The DNN model is then trained using the BP algorithm before being validated on a small test data set. This trained model can be applied to new samples, which must also be pre-processed using the same techniques and parameters that were used for training data.

Figure 11 – DNN Training and Application Pipeline.



Source: Author

Deep Neural Networks are powerful tools with a wide range of applications. In our work, we used them as a surrogate model in our Genetic Algorithm flow to propose a potential set of LSPR-based biosensors. The following section will discuss how the three main concepts, DNN, GA, and LSPR Biosensor, are discussed and approached in recent published research.

3 LITERATURE REVIEW

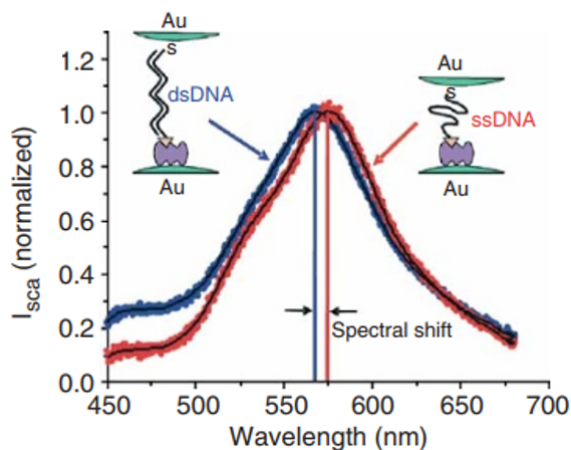
This chapter examines some of the most important works published in the field of optical biosensor devices in order to provide an overview of the significance of LSPR-based recent research. Our objectives are to highlight the motivations for our research while also demonstrating the opportunities that remain in the area. Furthermore, we discuss how important studies that used computational intelligence to improve design and sensor performance can serve as guidance and motivation for our work.

3.1 State of the Art on LSPR-Based Biosensor

Although the interest in plasmonic-based sensors has grown exponentially in the last decade, important journals have published relevant advances in the plasmonic field since the turn of the century. (SÖNNICHSEN *et al.*, 2005) described a project for an intramolecular rule capable of monitoring 70nm separation between molecules for over 3000 seconds.

The sensitivity of bio molecules to external influences is demonstrated by changes in their aminoacid structures. Förster Resonance Energy Transfer could be used to measure these changes (FRET). However, using FRET exposes the measures to the floating amplitudes of the evaluated signals, limiting the time window for observation. As a result, Sönnichsen proposes evaluating bio molecules aggregation and measuring the distance between noble metal nanoparticles using electron resonance in the visible spectrum.

Figure 12 – DNA Variation Measure.



Source: (SÖNNICHSEN *et al.*, 2005)

This phenomenon application is similar to what has been proposed using an LSPR-based biosensor, in which the different changes in the biological sample caused by the presence of a specific protein can be measured by a shift in the resonance curve. Figure 12 shows how Sönnichsen can quantify DNA modification using $\Delta\lambda$ quantization. According to (ANKER *et al.*, 2010), the sensitivity of resonance-based measures can be improved by determining the best combination of device geometrical aspects.

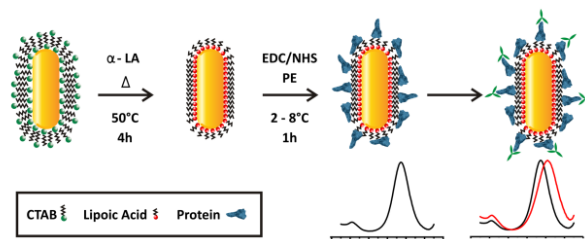
It is also difficult to accurately measure $\Delta\lambda$. (DAHLIN *et al.*, 2006) proposed an important evolution in this regard. His work enabled what is known as High Resolution Spectroscopy, which allows for measurements with sensitivities for $\Delta\lambda$ as low as $5 \times 10^{-4}nm$.

In general, Dahlin and his colleagues built a structure made up of an array of photodiodes that reduced noise sensitivity while increasing light absorption in the measuring equipment. This technique allowed for more precise detection of the surface resonance peak.

Some works, such as (OH *et al.*, 2014), present the application of LSPR for immunologic analysis in living beings via blood samples. In general, the author proposes measuring cellular cytosine to quantify the $TNF - \alpha$ parameter (tumor necrosis factor), which indicates the level of cellular inflation in the body. They propose integrating the LSPR-based biosensor into an optical-fluid platform to capture and stimulate immune cells with light excitation while checking the $\Delta\lambda$ for the presence of cytosine.

Still in the medical field, (VERSIANI *et al.*, 2020) from the Federal University of Minas Gerais recently published preliminary results of using a structure very similar to the one we optimized in our work to detect the presence of Dengue virus in human blood.

Figure 13 – LSPR-based Detection Pipeline.



Source: (VERSIANI *et al.*, 2020)

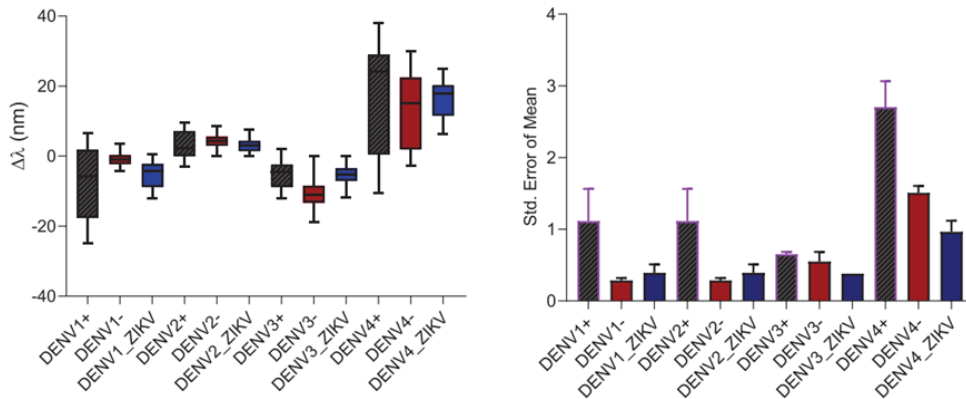
Figure 13 shows a portion of their proposed structure and analyte detection process, which makes use of the LSPR effect. The protein representing the target analyte is shown in blue, and one of the gold nanoparticles used is shown in yellow. The resonance curves show the

variation in the refractive index caused by the presence of the analyte, which red shifted its peak.

This work was extremely relevant, and (MACHADO *et al.*, 2022) published additional related results of the same research. They go into great detail about the diagnosis in this paper, but they also propose a computational systematic measurement system for quantifying $\Delta\lambda$ in order to show relevant genetic characteristics of Dengue disease.

Figure 14 depicts the graphical outcomes of their computational approach to experimental data collection. They first calculated $\Delta\lambda$ for the various serotypes of Dengue virus, namely DENV1, DENV2, DENV3, and DENV4, and then presented them in a novel way using Box-plots comparing seropositive to seronegative patients.

Figure 14 – LSPR-based Detection Pipeline.



Source: (MACHADO *et al.*, 2022)

They could use this to demonstrate the chaotic system that results from the diverse virus serotypes. It was also easy to show the genetic differences between each dengue type. Furthermore, they were able to use their biofunctionalized LSPR-based system to provide correct diagnosis with greater than 50% accuracy.

These two recent works, (VERSIANI *et al.*, 2020) and (MACHADO *et al.*, 2022) demonstrated the high relevance of LSPR-based systems; moreover, they demonstrated the importance of mixing the detection system with computational tools; and, despite their promising results, they affirm that optimizing the physical structure, that is, sensor geometry, is a critical step in improving disease diagnosis. As a result, the following section discusses how Artificial Intelligence is being used to optimize optical systems.

3.2 AI-Powered Biosensor Design

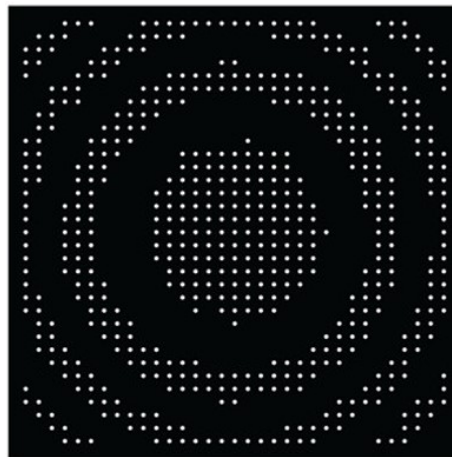
As we discussed in section 2.1, designing nanophotonic devices requires a significant amount of numerical simulation, especially when a specific optical behavior is expected to be achieved. This is known as the inverse design problem, in which the designer seeks to find a geometry that fits a specific resonance curve. However, achieving this geometry through estimation and numerical simulations is a time-consuming and difficult process due to the fact that this shape is not always intuitive.

As a result, the application of machine learning and optimization methods to inverse design problems is becoming increasingly popular. (YAO *et al.*, 2019) summarized the current techniques used to address these problems as genetic algorithms and gradient descent-based methods such as neural networks. Which is directly related to our work, that combines both worlds.

According to Yao, the main advantage of applying GAs to the inverse design problem is that it is capable of exploring a complex parameter search space and suggesting non intuitive geometries that would be very unlikely to be obtained by traditional methods from human designers.

Using an evolutionary approach, interesting works in the field of optical devices have been developed. (HUNTINGTON *et al.*, 2014) delivered one of them. They proposed a new opto-material made of a thin golden film with a novel shape called Subwavelength Lattice. They combined a matrix of 33x33 open or closed holes, similar to the one shown in figure 15.

Figure 15 – Subwavelength Lattice.



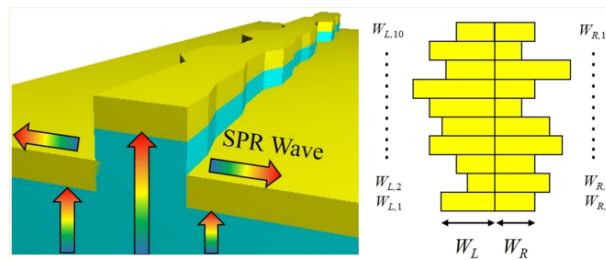
Source: (HUNTINGTON *et al.*, 2014)

It means that their structures have 1089 genes and a total of 2^{1089} different combinations. The diffraction index is used to assess the performance of their device. In a brute-force numerical analysis, it is not reasonable to exhaust all possible combinations. Instead, they created a model that could boost their performance index by 36.6 percent.

Proposing an evolutionary approach based on GAs that begins with a 600-member random population, after 210 generations, they were able to meet their expected threshold for a high-quality device. They did not specify the number of new offspring per generation, but even if they created a new entire population in each generation, they would try 126600 new devices, which is 321 orders lower than the entire search space. This work demonstrates how effective the evolutionary approach is for complex design processes.

Another relevant work was proposed by (FU *et al.*, 2018) with a method for optimizing a Gold nano-sized SPR biosensor using a micro genetic algorithm in the field of Surface Plasmon Resonance. Figure 16 depicts their structure.

Figure 16 – SPR-based Optimized Geometry.

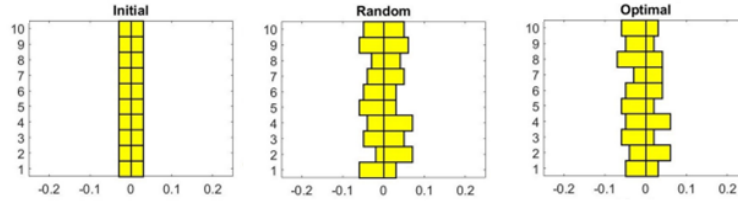


Source: (FU *et al.*, 2018)

It is composed of a series of ten nano rectangles. Their GA aims to find an optimal combination of this rectangle's shape. After 100 generations of 5 individuals each, they achieved an impressive 174 percent improvement in their fitness result by varying only the rectangle's length and keeping depth and height constant.

This enhancement was caused by the intensity of the Localized Surface Plasmon Resonance, which surged in the new rectangle edges. The original structure, one random structure, and the optimized structure are all shown in figure 17. It is obvious that the optimized one is completely counter-intuitive but significantly more efficient.,

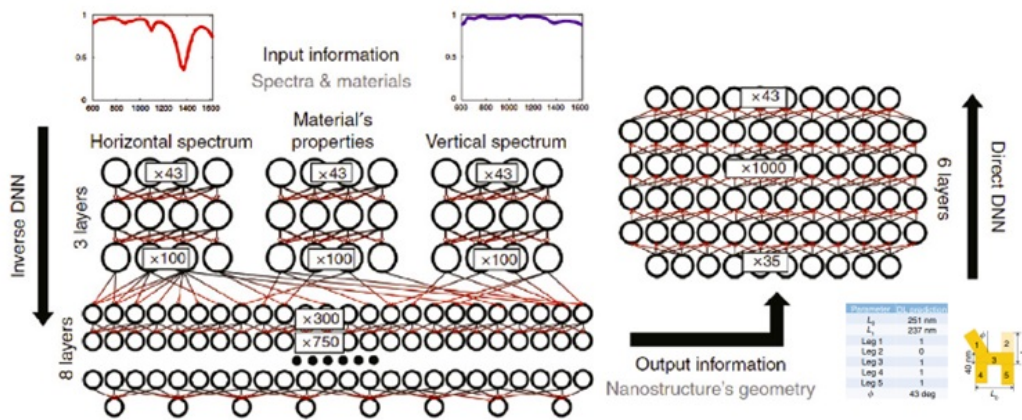
Figure 17 – SPR Different Geometries.

Source: (FU *et al.*, 2018)

Both works demonstrate how promising the evolutionary approach is, as well as how the cost function used for their methods is dependent on 3D numerical simulations. Some works have been developed proposing the use of Deep Neural Networks to model device optical behavior in order to mitigate the time and resource-consuming issues that result from those simulations.

(MALKIEL *et al.*, 2018) presented an important research study. Figure 18 depicts their proposal for a bidirectional complex Deep Neural Network. This structure initially has three parallel DNNs, each of which receives different parameters of the desired device as input. The first are 43 points of the horizontal resonance spectrum, the second are device materials, and the third are 43 points of the vertical resonance spectrum. After three layers, the outputs of these three DNNs serve as inputs for a larger fully connected eight layer DNN. The device geometry is predicted by this first DNN architecture.

Figure 18 – Makiel's Bidirectional DNNs.

Source: (MALKIEL *et al.*, 2018)

The device geometry parameters are then used as input for the second DNN structure,

which predicts the resonance spectrum and provides information for updating the network weights. Their sensor has an H-shape and 8 variable parameters, requiring a training dataset of approximately 16,500 samples generated through numerical simulation.

As a result of their approach, they were able to create optimal metasurface-based optical elements that are also useful for biosensing and analytes detection.

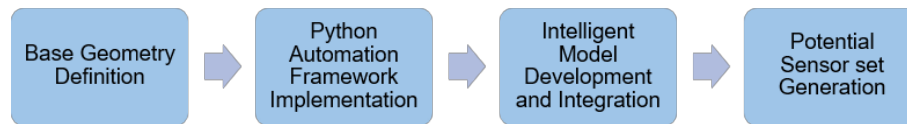
This literature review revealed vital information such as how GAs can be configured and DNN can be built to address the inverse design problem in nanophotonics. It also demonstrated that, while promising results as those presented by (MACHADO *et al.*, 2022) have been archived in the biosensing scenario, there is still room for improvement on LSPR-based biosensors, highlighting the relevance and serving as motivation for the research we are currently developing.

Therefore, the next section entails the entire development path that we took to propose our intelligent design method and to achieve a set of potential new devices.

4 METHODOLOGY

In this project, we proposed and implemented a systematic pipeline for developing a set of potential geometries of LSPR sensors with an optimized optical response. The flow chart in figure 19 shows the stages of this pipeline.

Figure 19 – Project Pipeline



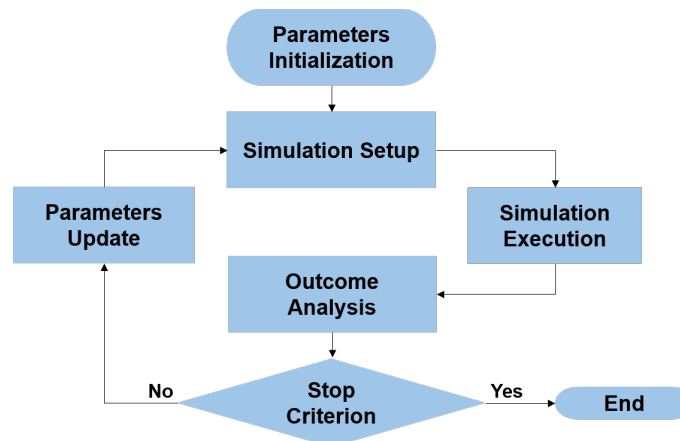
Source: Author

Each major phase of the design automation process and intelligent model development is described in this chapter. We begin by explaining the sensor design flow and how the base geometry was defined, going over the various aspects of Python automation framework implementation and concluding with an explanation of how the intelligent model was created and applied to the project to generate a set of potential geometries.

4.1 Sensor Design Flow

The sensor design flow consists of five major steps that are performed in a cyclic fashion and requires a stop criterion, which is typically a device performance metric.

Figure 20 – Design flow diagram.



Source: Author

Figure 20 diagram depicts these steps. The term "parameter" refers to both sensor geometry and simulation setup.

1. Parameters Initialization: The first step is to establish the starting point for device materials and geometry. It's also crucial to specify the type of simulation that will be run and the nature of the results that will be collected. There is a list of parameters that must be defined for this study:

- **Device materials:** includes the metal NPs material to be used, the substrate on which the NPs will be deposited, and the cladding material that represents the base biological sample.
- **Device geometry:** refers to the relevant dimension of each component on the sensor structure.
- **Mesh precision:** As the mesh dimensions become more precise, the simulation becomes more time and computer resource intensive. It is critical to find a boundary to optimize the choice of this parameter in order to ensure good simulation precision and optimal resource usage.
- **Simulation type and parameters:** When the basic geometry of a device is defined, it is a representation of the device's unitary cell. There are two main types of simulation, PML, and Periodic. Periodic denotes that the unitary cell will propagate and generate an array of unitary cells that will interact with one another, whereas PML denotes that the unitary cell will be simulated as a single cell. This work focuses on periodic simulations.
- **Light Source:** The light source will be critical in the simulation because it is responsible for stimulating the optical response of the design. A plane wave propagating in the Z direction across the device surface was used for our experiments.
- **Results monitors:** Lumerical employs a large number of monitors to collect simulation results. For this project, bi-dimensional monitors are combined to form a transmission box, allowing absorption rates to be measured on the fly.

All of the above parameters must be defined and validated before beginning the design project cycle.

2. Parameter Configuration: This is the first stage of the cycle, which means that after the initial definition and parameter initialization, each cycle will begin with new parameter configuration. Setting up the parameters states for changing device geometry after all other simulation specifics and sensor material have been kept constant for each different geometry trial.

3. Simulation Execution: Simulating an LSPR device entails measuring its optical properties and observing its behavior. The most important aspect of this stage is defining the simulation precision. We classified these precision based on two simulation aspects:

- **Mesh Step:** This parameter specifies how many points on the device's surface will be considered during numerical analysis.
- **Auto Shut-off Level:** It specifies the magnitude of the numerical error accepted as a simulation stop criterion.

We proposed two levels of precision combining those two aspects:

- **High Precision:** 3nm *Mesh Step* and 10^{-5} *Auto Shut-Off* level.
- **Low Precision:** 5nm *Mesh Step* and 10^{-4} *Auto Shut-Off* level.

Low precision setups were used for pipeline stages that required a large amount of data, whereas high precision simulations were used at the end of the experiments pipeline to validate a set of high potential geometries.

4. Outcome Analysis: This step entails gathering simulation data from the monitors and determining whether it meets the stop criterion for the objective metrics. This analysis could be done graphically through a software interface or numerically through software integration.

In our project, we propose analyzing the resonance curve for the LSPR structure. The goal is to determine the peak of this curve because the higher the peak, the greater the chance of greater sensor precision.

So far, we are not evaluating curves bandwidth which we expected to be as narrow as possible. This is currently a limitation that will be address in future work.

5. Parameters Update: When the most recent trial results do not meet the stop criterion, a new geometry must be proposed and tested so that the cycle can continue until it meets the criterion. This new geometry could be proposed by an intelligent method, which is the goal of this work.

4.2 Base Geometry Development

The process for defining sensor base aspects followed the cycle outlined in the previous section, but each step was completed by hand, with no automation. The goal was to come up with a geometry that had a minimum satisfying absorption level over a wide range of wavelengths.

The first arbitrary decision was to shape gold nanoparticles as golden nanodisks. The gold disks were first deposited on a silicon substrate, with the cladding representing water's refractive index.

The radius and thickness of the disk are the relevant geometrical properties. The distance between each unitary cell on the periodic array is also of interest because it affects the intensity of the plasmonic coupling. As a result, the initial set of parameters that could be improved included:

- Disk Radius
- Disk Thickness
- Distance Between Disks

However, two key findings emerged from a series of preliminary experiments:

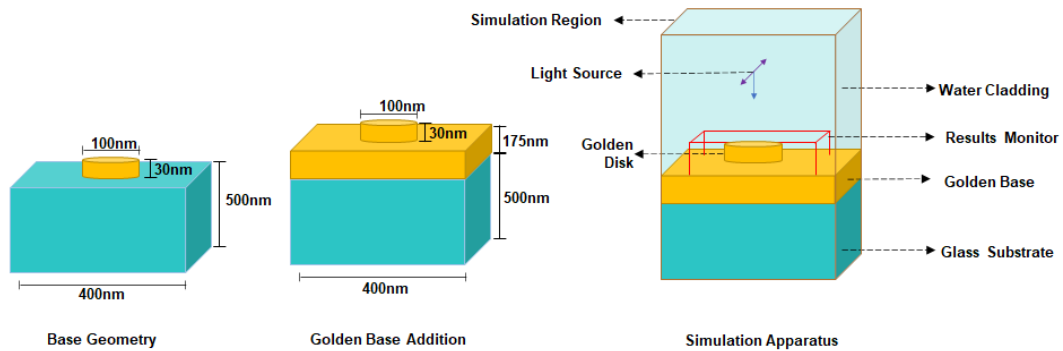
1. Disk thickness has a major influence on the wavelength where the absorption peak occurs rather than on its intensity. So, we could anchor this parameter in a range of desirable wavelength values.
2. To facilitate the fabrication process and improve evanescence field enclosure into the device surface, a golden rectangular film could be inserted between silicon and gold NPs.

As a result, the final set of relevant and changeable parameters is as follows:

- Disk Radius
- Distance Between Disks
- Golden Base Thickness

Therefore, an example of the base geometry, the new geometry with the golden base and the simulation apparatus is given in figure 21.

Figure 21 – Sensor Base Geometry and Simulation Apparatus.

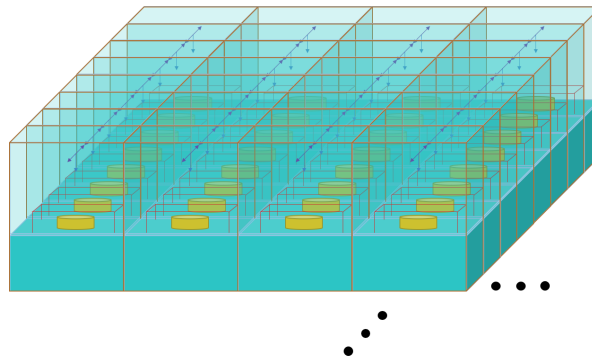


Source: Author

The brown box defines simulation and structure limits for the unitary cell. The red box is the results collector monitor and the axis represents the light source. Blue rectangle is the substrate and the golden disk is the gold NP.

Furthermore, keep in mind that figure 21 depicts a unitary cell of the entire device. Figure 22 depicts the periodic arrangement to provide a better understanding of the overall structure.

Figure 22 – Base Geometry Periodic Representation.



Source: Author

The periodic arrangement allows interaction between each of the nanodisks, and achieving a good range of the parameter "Distance Between Disks" means that the plasmonic coupling[2.1] will be optimized and the resonance intensity will be increased, thereby improving sensor sensitivity.

It is important to note that the entire process of obtaining the base geometry took approximately 3 months, which is a reasonable amount of time. The manual execution of the

design flow is a key factor in this time-consuming process, which led to the need for automating the process, allowing us to generate a large number of possible geometries in a short amount of time.

4.3 Automation Framework Development

To automate the design flow process and make it simple to insert intelligent parameter search space, we created an object-oriented integration framework based on two main technologies.

1. **Python:** a widely used and broadly shared programming language that includes a large number of open-source data science libraries for data manipulation and analysis. If coupled with good programming guidelines and a class-oriented structure, using Python is a good way to provide an easy-to-maintain framework.
2. **LumApi¹:** an Application Programming Interface (API) that connects Ansys Lumerical to third-party software frameworks. This is a work-in-progress API that already allows users to access the core features of Lumerical software and includes a Python version for easier integration with the programming language we used.

The library we created to provide framework methods is called *lsprlib*. It is intended to give the user the ability to execute all steps of the design flow while also giving them the flexibility to do so in a variety of modes. It allows users to write scripts that use the *lsprlib* classes to perform a variety of experiments based on a range of values for relevant parameters and simulation setups. The major experiment modes available are:

- **Individual:** For a single sensor morphology, the user provides simulation setups and geometry boundaries, then simulates and displays graphical results.
- **Parametric Sweep:** The user specifies an interval for relevant parameters and variation steps, and the framework runs the simulation cycle until all combinations are exhausted. By the end it provides a CSV file containing all relevant sensor and optical results data.
- **Random Generation:** The user specifies the parameters' limits, a random seed, and the number of individuals. The framework will generate this many random geometries, run simulations, and report the results in a CSV file.
- **Intelligent Model:** The user specifies the parameters' limits, a random seed for an initial sensor population, the size of this population, and the number of cycles to run. The

¹ <https://support.lumerical.com/hc/en-us/articles/360037824513-Python-API-overview>

framework will propose a set of potential geometries with good optical behavior using an intelligent model based on a Genetic Algorithm and a Neural Network surrogate model. More information about the intelligent model will be provided in the following section.

The framework is composed of the following class structure to serve as a tool for project design flow:

Simulation: This class represents one cycle of design flow performed by a single new geometry. It contains all the methods that interact with the Lumerical API and stores simulation results for the sensor being tested.

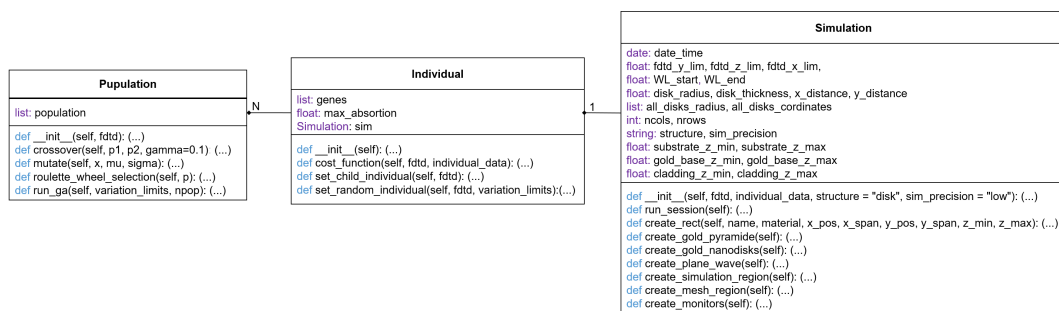
Individual: This class represents a single sensor and has three attributes. One of them is *sim*, which is an instance of the Simulation class. It also includes methods for generating relevant parameter values or establishing new values based on intelligent model recommendations. We named these relevant parameters *genes* because they represent the distinct characteristics of a new sensor.

If user wants to generate a random device, this class has *set_random_individual* method. When using intelligent mode design, the model will recommend the parameters, and this class will use the *set_child_individual* method to set up the individual genes. Each individual has a unique absorption peak value, which is referred to as the *max_absortion* attribute.

Population: This class encapsulates the whole set of Individual instances generated and promotes the initialization and evolution of a population. This individual generation can be based on both a random approach or an evolutionary approach.

The following diagram depicts how those three classes interact with one another. Population has a list of N Individuals and one Simulation instance exists for each individual object.

Figure 23 – Framework Class Diagram.

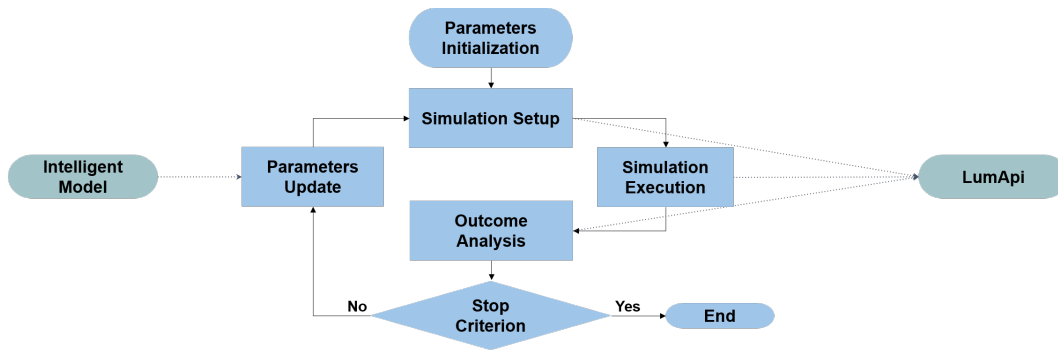


Source: Author

The framework also includes a DNN class, which can be used independently of the other three and represents a neural network model that states the optimizable cost function of the evolutionary approach. We trained this model before beginning a new population evolution, its primary goal is to replace the majority of numerical trials as the primary component of individual testing.

Therefore, the flow diagram on figure 20 was updated based on the framework features and its new version is shown here:

Figure 24 – Complete Design Flow Diagram.



Source: Author

Next session discusses how DNN and GA interact with each other as part of the intelligent model development.

4.4 Intelligent Model Development

The intelligent model proposed here combines two commonly used artificial intelligence techniques: the Genetic Algorithm for parameter optimization and the Neural Network to replace numerical simulations on the GA optimization function.

4.4.1 Genetic Algorithm

As we stated in subsection 2.3, GAs are composed by four major stages: selection, cross-over, mutation and fitness calculation. Following that, we discuss how each stage was implemented in *lsprlib*.

Selection: We compared two main major selection methods, namely K-way tournament, and rank selection, in a single GA flow of 50 initial individuals and 20 generation.

The goal was to comprehend method saturation and best parent priority. We anticipated that the roulette wheel wouldn't provide an efficient parent pickup because fitness values begin to become similar between individuals very quickly. The k-way tournament requires fine tuning of the k parameter, which was determined by calculating which k gave priority to the best 20% of parents while also giving the other 80% the option of being chosen.

Cross-Over: The individual's three variable genes, Radius, Distance, and Base Thickness, are continuous parameters. Therefore, we expected good results with linear cross-over, however, we also tested one-point cross over in order to validate our hypothesis.

As a result to define the arrangement of Selection and Cross-over methods we compared the following combinations:

1. Linear Cross Over and Rank Selection.
2. Linear Cross Over and K-Tournament Selection.
3. One Point Cross Over and Rank Selection.
4. One Point Cross Over and K-Tournament Selection.

Mutation: The mutation processes are employed on a small group of offspring, and their main purpose is to introduce new features into a generation. When the evolutionary problem is encoded by binary genes, swap procedures for mutation are often used, however this is not the scenario in our cases because we have a continuous numerical representation.

As a result, we chose to alter 20% of the children in a generation with a broad new value for one or more genes. This is a value picked at random between the changeable gene boundaries. For example, if the algorithm decides to alter the individual gene representing disk radius, it will be randomly assigned a new value between 40nm and 90nm.

Fitness Calculation: An optimum geometry is discovered by maximizing or minimizing its fitness. Our strategy proposes maximizing the peak of the absorption curve and identifying the individual with the most intense peak.

The absorption peak is the result of a numerical simulation, but in our approach, those simulations are modeled with a Prediction model, so the DNN surrogate model represents the fitness function to be optimized. That is, each new individual fitness generated by the GA have its fitness predicted by the DNN. Only a small group of best individuals will be re validated by numerical approach. The following section describes the methodology for developing and implementing this surrogate model.

4.4.2 NN Surrogate Model

The Deep Neural Network model was created with the following principle in mind: it is a surrogate model, and the best recommended structures will be validated later in numerical simulations. That is, its primary goal is to provide precise direction to the results, but it is not expected to be almost 100% exact.

As a result, it was critical to establish a fair balance between training data set length and model accuracy. Less numerical simulations are required to create input data when the training data set is smaller. Furthermore, because the model's goal is to exactly reduce design flow simulation costs, training should not be time or resource intensive.

The DNN development pipeline consists of three major steps: data pre-processing, model training, and model validation. These three processes were repeated for different amounts of training data before being validated in the brand new validation data set.

Data Pre-processing: The input data includes the three relevant parameters listed in 4.2, namely the disk radius, disk distance, and golden base thickness. The output data, on the other hand, is a prediction of the absorption curve peak. However, the three input attributes have a different order of magnitude, therefore we used a normalization method to give them all the same weight on the neurons training. We saved the bounds, X_{min} and X_{max} for each attribute after normalizing the data. This is significant since our new samples must be normalized within the same value range. This format is comparable to the leave-one-out validation format.

Architecture Definition: The DNN model has four main parameters: the number of hidden layers, the number of neurons per layer, the learning rate, and the layer activation function. We defined these parameters using the grid search technique for layers and neurons with the following ranges and steps:

- **Number of Hidden Layers:** 1 - 10, step of 1;
- **Neurons per Layer:** 5 - 100, step of 5;

The early stop approach was used to prevent this grid search from getting too costly. We created a validation set of 5% with a patience parameter of 15 epochs and a validation loss delta of 0. It means that during the training process, the model will be trained with 95% of the data and validated with the remaining 25%, but the training will be ended when the validation loss stops changing significantly.

The learning rate must be tailored to the problem at hand; however, literature suggests that 0.0003 is a good starting point for avoiding saturation during the learning process while also

avoiding it taking too long to complete. We performed the layers and neurons grid search with this value, evaluated the results, and then tested new learning rate values of 0.03 and 0.003 for the best architectures.

We also defined the activation function based on a review of the literature, which revealed that the majority of DNN models currently use the ReLU activation function. It also fits better with our project because of the lower computing cost, as it does not update all neurons for each back propagation iteration, reducing training cost.

Finally, the model was tested on a completely new test data set generated by numerical simulation. Our test metric was the percentage difference between the model prediction and the expected result, shown in equation 4.1. The smaller this proportion, the more accurate the model.

$$F = \frac{1}{N} \sum_{i=1}^N \left| 1 - \frac{y_i}{\hat{y}_i} \right| \quad (4.1)$$

At the end of this stage, we compared the best DNN architecture results for various training input sizes. It is significant because if less numerical simulation is required, and the DNN can be trained with a small portion of the training data set and still be satisfactory efficient, we could propose a less costly Intelligent Model (IM). The following section describes the IM application flow.

4.5 Intelligent Model Application

Following the implementation of the intelligent model, we are able to run a complete experimental flow, from scratch to a final set of high potential geometries. We divide the experiments into four major phases, which are summarized in the table 1.

Table 1 – Experimental Flow Summary

	Step 1	Step 2	Step 3	Step 4
Phase 1: Data Generation	Define parameters range	Generate random data	Evaluate parameter search space	-
Phase 2: NN Model Training	Define NN parameters	NN train	NN validation	NN model saving
Phase 3: GA Application	Generate new random initial population	Create new generation	Validate 10% best individuals in numerical simulation.	Repete steps 2 and 3 for N generations
Phase 4: Final Set Validation	Final set high precision numerical simulation	Parameters tolerance analysis	-	-

Phase 1 is in charge of the data generation step for both training the Neural Network Model and visualizing the search space. Evaluating the search spaces allows for a more accurate definition of the parameter range in posterior random generations.

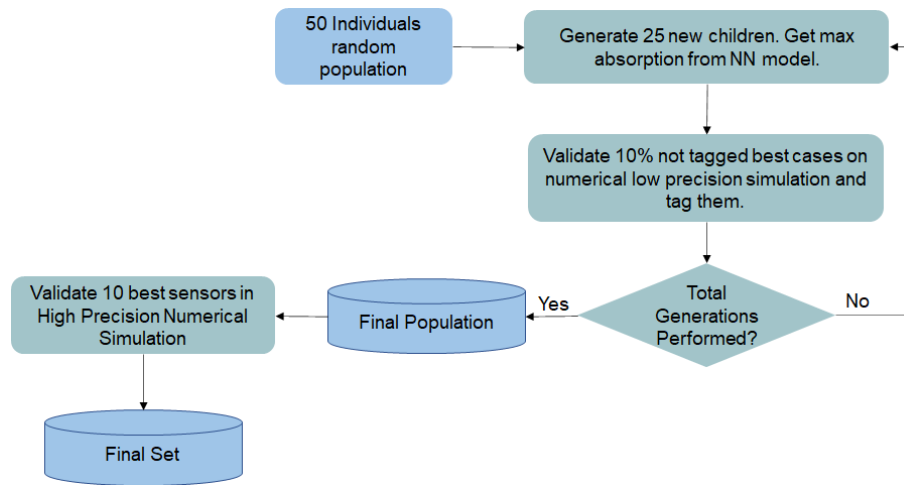
Phase 2 is when we implemented the Neural Network Model based on the information in section 4.4. We used the grid search technique to define DNN depth and neurons per layer number, then trained the chosen architecture with all available data and validated it in a completely new data set to measure the mean error between NN and numerical models.

Phase 3 is the GA application procedure. We begin by defining an initial random population as well as the number of generations and children per generation to be performed. For each new child in a generation, we first evaluate its cost function using the DNN model, then we take the 10% better individuals and test them on numerical simulation in low precision mode, tagging this individual with a flag to avoid testing them again in the next generation if they remain in those 10%.

In our project, we have 50 sensors in the initial generation, 25 children per new generation, and 25 new generations, for a total of 700 individuals. The combination of numerical simulation and DNN validation is critical for ensuring quality and efficiency. In the worst-case scenario, we would have 70 numerical simulations instead of 700. However, we anticipate a much lower number of numerical trials, because when evolution begins to saturate, the majority of the best individuals remains the same and we only re validate new top-10% individuals in each generation.

Phase 4 is when we validate the final 10 best cases in a high precision mode numerical simulation to obtain the most accurate computational optical metrics possible. Figure 25 depicts the pipeline combinations of phases 3 and 4.

Figure 25 – GA Application Procedure Pipeline.



Source: Author

The results for each of the geometries obtaining processes are specified and discussed in the following chapter: base geometry, automatically generated geometries, and optimized geometries

5 RESULTS AND DISCUSSION

In this chapter, we will discuss the outcomes of our research as illustrated in figure x. We began by characterizing the base geometry, which served as the starting point for the entire development. Once we had the base geometry and the automation framework in place, we were able to generate new geometries using both parametric sweep and random search methods. The final step was to use the intelligent model to implement the systematic sensor generation flow. We also show some fabrication results for one of the first high-sensitivity geometries created with the automated flow.

Figure 26 – Results Pipeline.



Source: Author

5.1 Manually Obtained Base Geometry

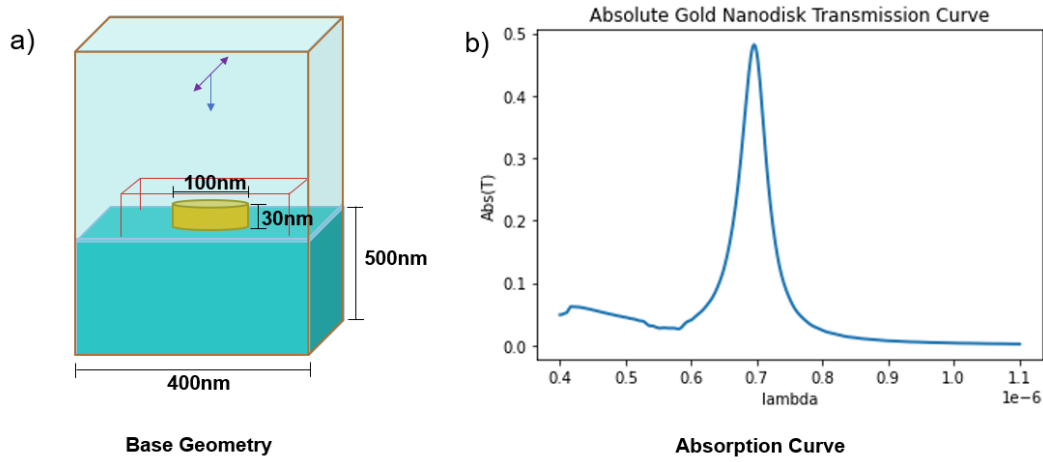
The base geometry was obtained after two months of manual execution of the sensor design flow[4.1]. This preliminary study was carried out prior to our project by other members of the research group. The nanoparticle material was arbitrarily chosen, and it is made up of gold nanodisks deposited on a glass substrate. The first study looked at three parameters:

- Disk Radius
- Disk Thickness
- Distance Between Disks

It was intends to accomplish a geometry with a single resonance peak nearing 50% intensity. After two months of manual tries, it was possible to achieve the initial milestone. The best parameters are described below, along with figure 27, which depicts the base geometry and its absorption curve.

- Disk Radius = 50nm
- Disk Thickness = 30nm
- Distance Between Disks = 400nm

Figure 27 – Base Geometry and Absorption Curve.



Source: Author

It is important to note the quality of the absorption peak, which has a distinct peak centered at 700nm. This initial geometry is critical for establishing the operation wavelength region and parameter limits. As a result, the following section demonstrates how we used the automated Python framework to search for a better combination of device parameters and how much we improved the absorption peak.

5.2 Automatically Obtained Geometries

In the mean time between defining the base geometry and beginning to use the automation framework, we decided to include a fourth parameter, a golden film between the nanodisks and the substrate. This golden film was supposed to boost absorption from free electrons on its surface. Furthermore, we decided to anchor the golden disk thickness in order to keep the device's operation region near to 700nm. As a result, we will now refer to the changeable parameters as:

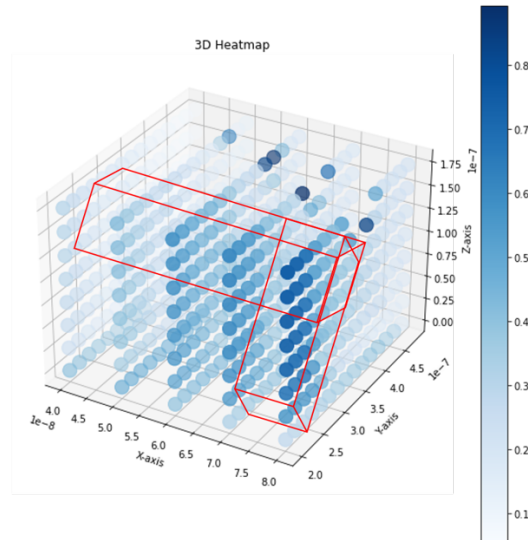
- Disk Radius
- Distance Between Disks
- Golden Base Thickness

With the availability of the automation framework we were able to execute two important experiments. The first one was a parametric sweep varying the relevant parameters in a linear range described as follow:

- Disk Radius: 40 to 80nm, step of 10nm.

- Distance Between Disks: 200 to 500nm, step of 10nm.
- Golden Base Thickness: 0 to 200nm, step of 5nm.

Figure 28 – Parametric Sweep Heat Map.

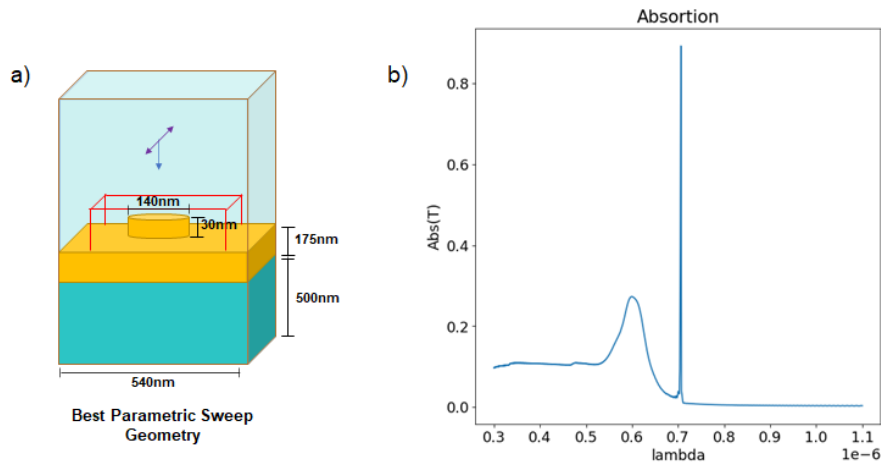


Source: Author

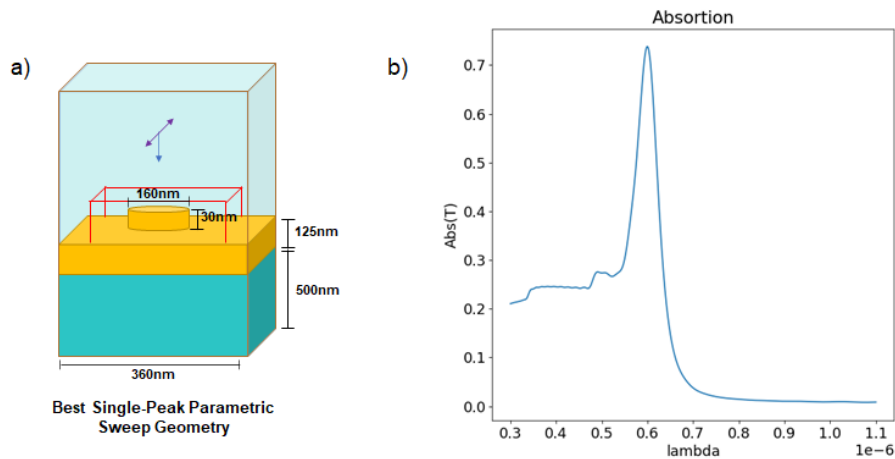
The entire experiment was executed with low precision simulation parameters, it took 32 hours to generate 594 valid new geometries. A geometry is considered valid when the numerical simulation ends without any divergence error. The heat map in figure 28 shows how the search space was explored and where the best fitness stands. This was a high-value experiment and data set for future comparison with optimization search space methods like the genetic algorithm.

Figures 29 and 30 show the best two geometries obtained from the parametric sweep. First one with 89% absorption peak and the second with 73%. It is worth noting that the best one is not a single peak curve, but its highest peak is noticeably more intense than the first.

Figure 29 – Parametric Sweep Best Geometry.



Source: Author

Figure 30 – Parametric Sweep 2nd Best Geometry.

Source: Author

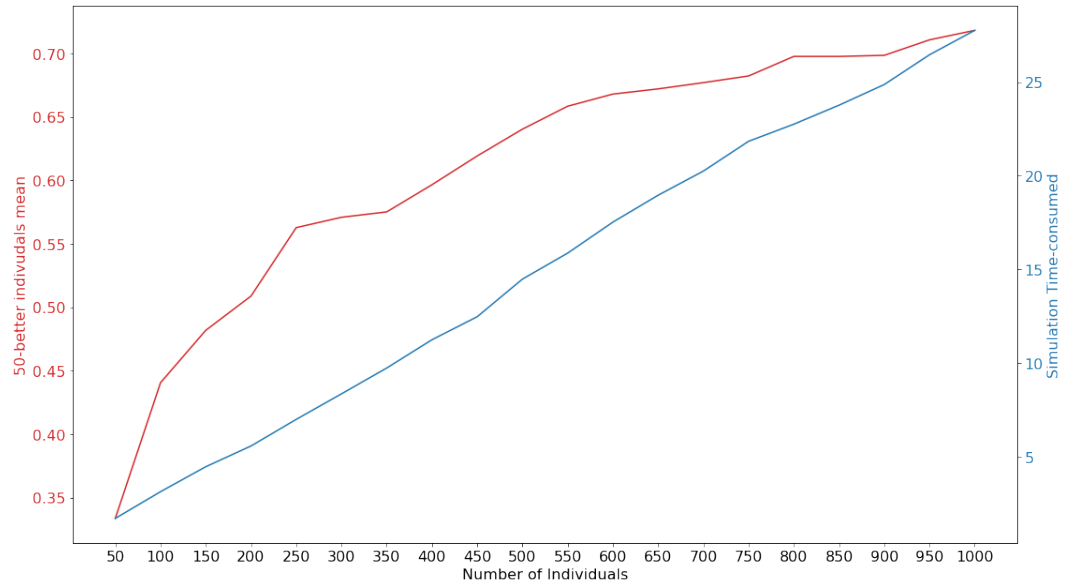
These two geometries were validated in high precision mode simulation, and their curves retained the same format, but with a decrease in their absorption's peak from 89 percent to 75 percent for the first case and from 73 percent to 69 percent for the second. This demonstrates that while the level of precision may be more relevant in some cases than others, the order of magnitude for both levels is very similar, which validates the approach of using low precision for huge amount of data and high precision for validations.

The second experiment involved conducting a random search within the same range as the parametric sweep. The goal was to calculate the mean absorption peak value for the top

50 individuals in a population of size ranging from 0 to 1000.

Figure 32 depicts the evolution of the 50-best individual mean, while the blue axis depicts the amount of time required to generate all individuals. It is worth noting that achieving a 70 percent 50-individual absorption required approximately 30 hours of numerical simulation.

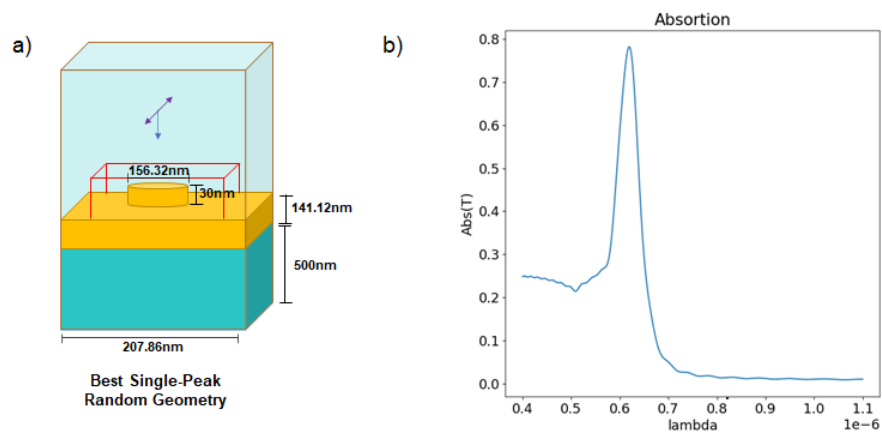
Figure 31 – Random Search Summary.



Source: Author

Figure 32 depicts the best single-peak case obtained through the random search.

Figure 32 – Best Single-Peak Random Geometry.



Source: Author

Having the automation framework ready to trigger simulations was clearly an im-

portant tool for improving the efficiency of the sensor design flow. Using the framework, it was possible to generate a set with a few individuals that performed significantly better than the base geometry, increasing the simulation results from nearly 50% to 75%. However, it became clear that the greater limitation is the time and resources required, as all experiments required more than a day to generate only a group of high potential geometries.

Therefore, in the following section, we present the results of combining a Genetic Algorithm with a Deep Neural Network in order to save time and resources while increasing the likelihood of generating a complete population of high-sensitivity geometries.

5.3 Smartly Obtained Optimized Geometries

This section describes the outcomes of the Intelligent Model's implementation and application . It first describes the experiments carried out to determine the best approach for the Intelligent Model (IM), specifically the Genetic Algorithm (GA) techniques used and the DNN architecture parameters. Finally, it compares the GA results for creating a new population to the previous results obtained without the IM.

We were able to generate 4700 random samples using the automation framework. These samples were used in a traditional grid search to determine the number of layers and neurons per layer. The relevant geometric parameters are fed into the DNN, and the output is the absorption curve's peak. We used a test data set of 83 brand-new samples. The loss function was shown in the equation 4.1. Table 2 summarizes the top ten architectures.

Table 2 – DNN Grid Search Summary

Layers	Neurons	Error
4	50	0.064435
5	80	0.065522
4	90	0.066007
2	80	0.067178
4	70	0.067281
5	90	0.067658
5	70	0.067983
4	60	0.068424
3	60	0.068891
3	80	0.072463

Following the grid search analysis, we define a DNN model with four layers and

fifty neurons per layer. We then evaluated model's sensitivity to the number of input samples. Since generating samples through numerical simulations is costly, the fewer training samples the model requires to achieve satisfactory accuracy, the better. Table 3 depicts the relationship between model error and number of training samples.

Table 3 – DNN Training Size Sensitivity

Number of Training Samples	Error
500	0.1044
1000	0.0936
1500	0.0887
2000	0.0845
2500	0.0891
3000	0.0835
3500	0.0756
4000	0.0701

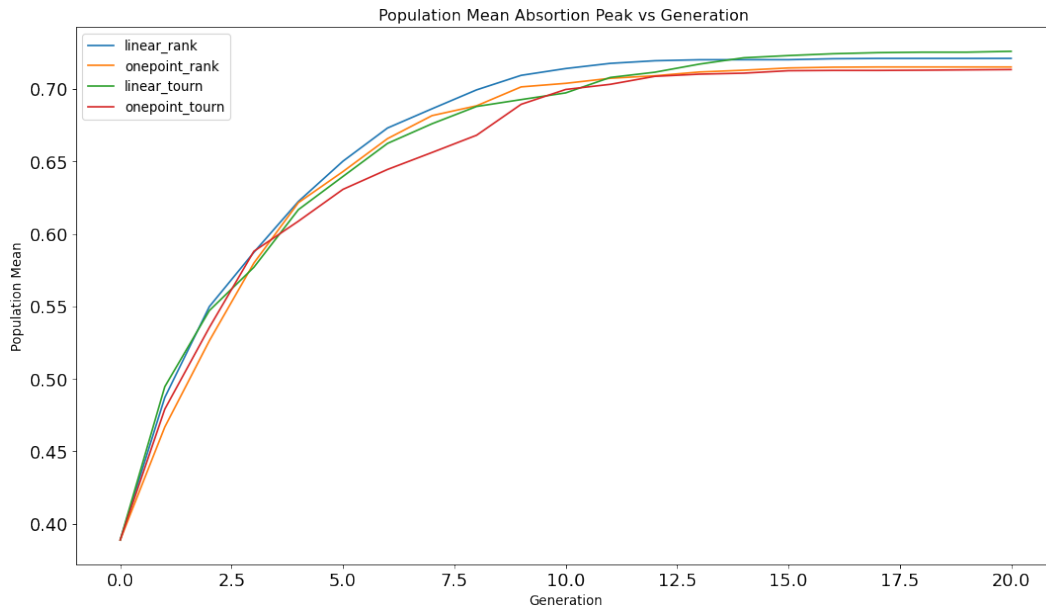
It is worth noting that the model's accuracy improves with the size of the input data set. However, as a surrogate model, and considering that the flow will validate the 10% best individuals for each generation, it is permissible to state that a data set with only 1000 samples is sufficient for training the model. As a result, our final approach includes a DNN with four hidden layers, 50 neurons per layer, and 1000 samples for training.

After defining the DNN, we validated the Genetic Algorithm using the DNN as the cost function and without performing numerical validations. By doing so, we could quickly compare the various combinations of selection and cross over methods. We compared the following combinations:

1. Linear Cross Over and Rank Selection.
2. Linear Cross Over and 3-Tournament Selection.
3. One Point Cross Over and Rank Selection.
4. One Point Cross Over and 3-Tournament Selection.

Figure 33 depicts the preliminary validation results. The four methods clearly had approximate behavior, but 2 and 4 had slower saturation and better final performance. The 3-Tournament selection performed better and had a better saturation level. Based on these studies, our final structure consists of Linear Cross Over and 3-Tournament Selection.

Figure 33 – GA Methods Outcome.



Source: Author

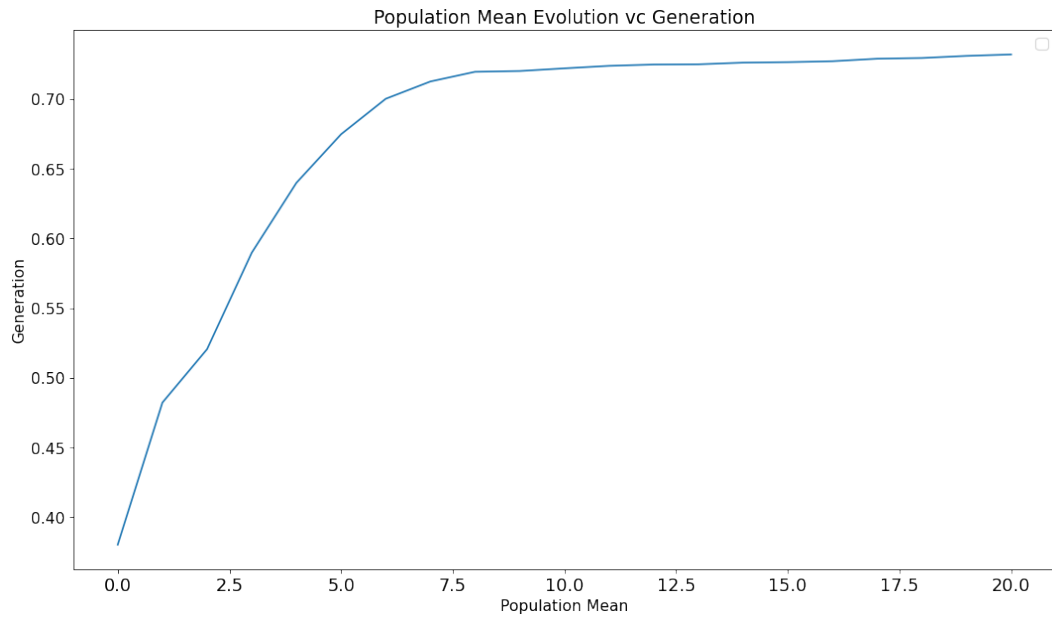
As a result, table 4 steps summarizes the IM final architecture.

Table 4 – Intelligent Model Final Architecture

	Applied Technique	Reason
DNN Model	4 Hidden Layers with 50 Neurons trained by 1000 samples.	Grid Search Validation.
Selection	3-K Tournament Selection	Allows for the best search space stimulus while still giving priority to the best parents.
Cross-over	Linear Cross Over	Genes are represented by continuos numerical data.
Mutation	Random Insertion	Not binary and diferent magnitue orders for sensor parameters
Fitness Calculation	DNN Surrogate Model	Avoid timing and resource intensive numerical simulations

Using the automation framework, which now includes the IM, we generated a random population of 50 people and triggered 25 generations of 25 new members each. Figure 34 depicts the evolution of the population mean with respect to generation number. It is worth noting that the method achieves convergence after 20 generations, with a population mean close to 73%.

Figure 34 – Final Genetic Algorithm Evolution.



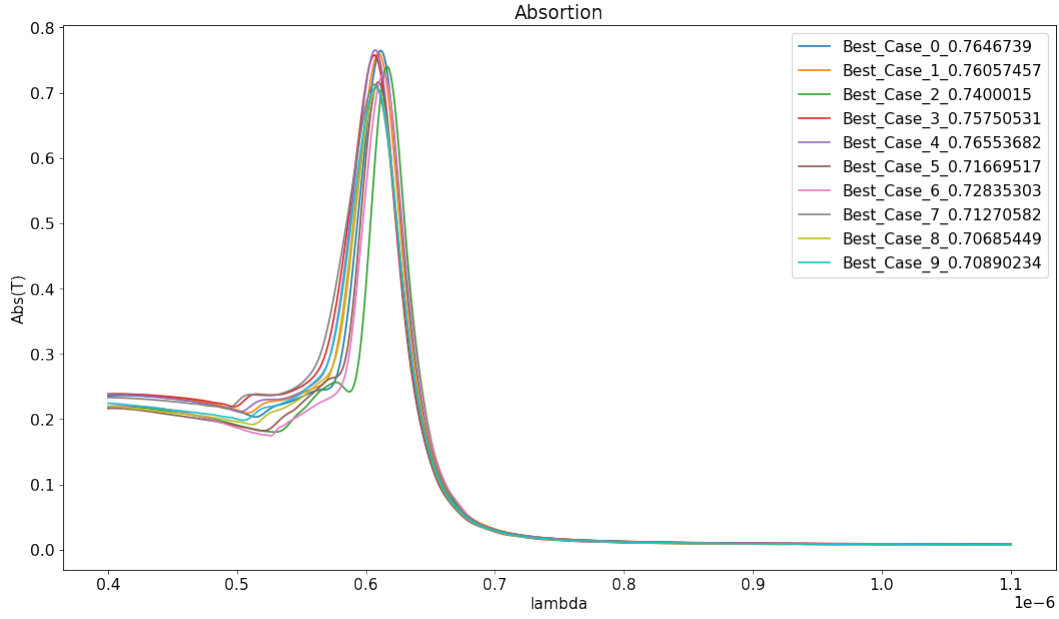
Source: Author

After validating the final set of 10 geometries and obtaining high precision simulation results, we achieve a mean error of 2.60% between the high precision and our flow results, which is expressively low and validates the entire process. Table 5 compares the results before and after final validation, and figure 35 shows the best cases absorption curves.

Table 5 – Final Set Summary

Disk Radius (nm)	Distance (nm)	Disk Thickness (nm)	Max Absorption	Max Absorption Valideted	Error
89.68	210.69	148.95	0.774152	0.764674	1.22%
88.95	207.52	155.29	0.769727	0.760575	1.19%
88.44	227.55	125.32	0.755423	0.740001	2.04%
86.55	204.93	153.66	0.742345	0.757505	2.04%
86.92	207.77	166.47	0.741672	0.765537	3.22%
84.55	226.93	153.19	0.741638	0.716695	3.36%
88.19	221.04	72.11	0.740552	0.728353	1.65%
83.91	206.44	159.20	0.739038	0.712706	3.56%
83.86	220.87	167.62	0.737214	0.706854	4.12%
83.72	216.30	166.72	0.735367	0.7089	3.60%

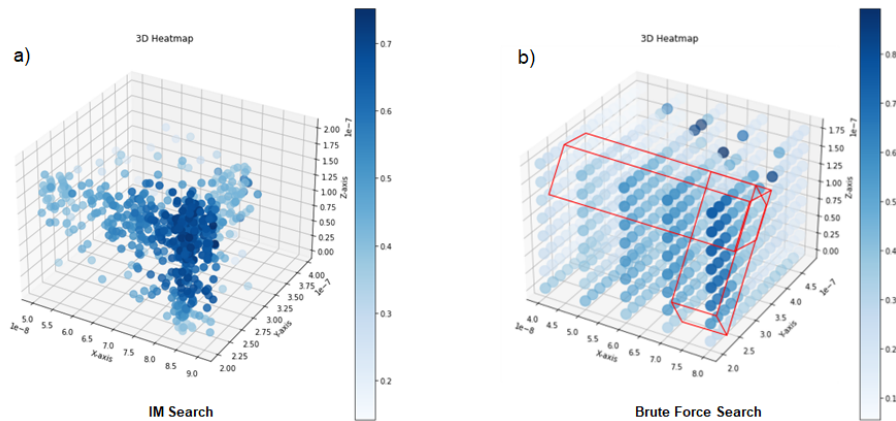
Figure 35 – Optimized Geometries Absorption Curves.



Source: Author

Finally, we compared brute-force search space exploration to GA search space exploration. Figure 36 shows that, while the method did not explore the entire space, it did explore a large portion of it and converged to the same region where the parametric sweep revealed its best individuals, which we interpret as the global maximum for our problem.

Figure 36 – Search Space Exploration.



Source: Author

Aside from reaching a high-sensitivity total population, another significant accomplishment is the time-consuming process. We need 1000 numerical samples to train the DNN for

the IM approach, but once trained, we can use it as a tool for all of the other experiments.

The final IM flow required 25 numerical validation simulations to achieve a 73% mean population. The random approach presented in figure 32 achieved 70.9% after 1000 simulations. However, it began to saturate in 850 individuals when the mean was 70.1%, implying that 150 numerical trials raised only 0.8 p.p, whereas using the IM approach, we could increase from 70.9 percent to 73% with only 35 new numerical experiments. The GA spent 3 hours total on all numerical validations.

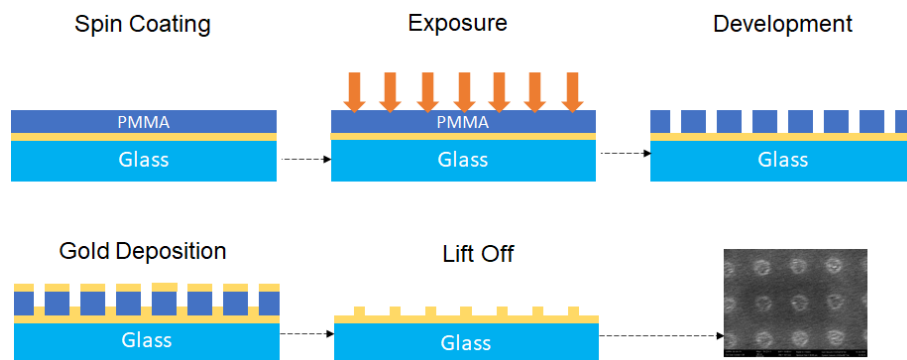
In conclusion, having an automation framework for random and parametric sweep is already a significant accomplishment that has resulted in significant device performance improvements and time savings when compared to manual base geometry definition. Furthermore, we could address the expensive simulations and random search saturation with the IM, which executes in significantly less time while still improving final device performance. The initial design had 47% maximum absorption, while the final best population has a mean of 73%, representing a 55% improvement.

All of the above results were obtained through computational means. The following section shows the fabrication results of one of the obtained geometries. We were unable to fabricate the final optimized geometries in time, so this is the first high performance obtained structure presented in figure 29.

5.4 Device Fabrication

A technique known as Electron Beam Lithography was used to create the nano device we proposed. The steps for this process are depicted in figure 37 .

Figure 37 – Fabrication Process.



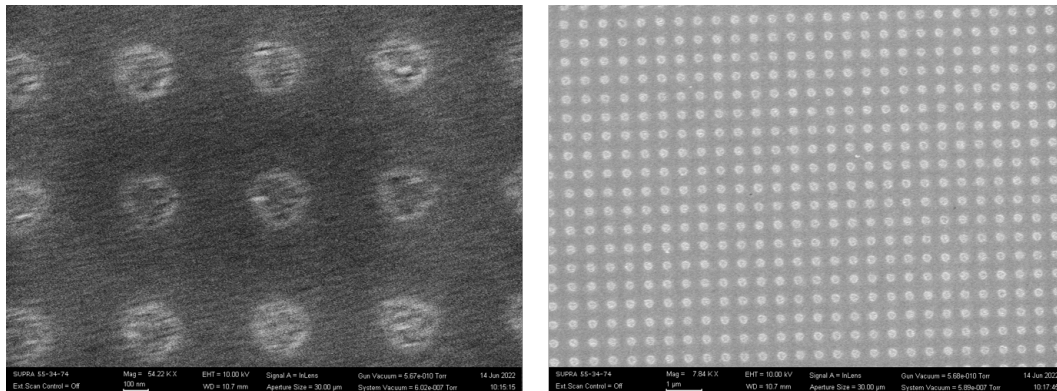
Source: Author

This procedure begins with the preparation of the glass substrate layer, followed by the deposition of an organic material layer over it, in this case (polymethyl methacrylate) PMMA.

The second stage involves scanning the PMMA layer with a very precise electron beam in the positions where the golden nano disks will later stand. This is known as exposure, and the structure will have a new shape with small holes for metal deposition as a result of these steps.

The metal, in this case gold, is then deposited over the PMMA and Glass layers, and the remaining PMMA material is lifted off, revealing only the material of interest, which are the golden base and nano disks. Figure x depicts the fabricated device with the periodic arrangement in two scales.

Figure 38 – Fabricated Device.



Source: Author

6 CONCLUSION

6.1 Accomplishments

The primary goal of this work was to create an Intelligent Model (IM) for designing plasmonic biosensors that combined a Genetic Algorithm (GA) and a Deep Neural Network (DNN). There were some milestones that needed to be met in order to achieve this primary goal.

The first was to build a framework to automate the entire computational design flow. This framework was built with Python and Lumerical API and was extremely useful for generating large amounts of data in a short amount of time while also providing automated outcome analysis.

With the automation framework in place, we could then develop each member of the IM. Initially, the DNN model could be developed and validated in a variety of ways, including its error level and sensitivity to training data set length. For example, we could build a model with a mean error of 9% for a small input data set of 1000 samples.

Using the trained DNN it was possible to implement the GA and validate it with different selection and crossover methods choosing the combination with better performance and space exploration, composed of Linear crossover and 3-way tournament selection.

By combining these intelligent methods, we were able to achieve a 73% absorption peak mean population of 50 different geometries in a more reasonable amount of time with an expressively small number of numerical simulations. When compared to the initial base geometry, this performance represented a 57% improvement.

Therefore, we can say that all milestones for the proposed objectives of this work have been met. However, there is still room and opportunity for evolution, which we will discuss further below.

6.2 Future Work

As previously discussed, LSPR-based biosensors have emerged as a promising technology for monitoring human health indicators. Their nano-shape and low-cost experimental apparatus allow for less expensive but accurate diagnosis. However, there are still many opportunities in the field, which motivates us to keep working in order to provide efficient devices.

As a result, while our study produced intriguing results, it was based on a simple geometry with only three variation parameters. Furthermore, our intelligent model only looks for a number that represents the peak of the absorption curve. We understand that a high sensitivity device can have much more unusual geometry and a more complicated resonance spectrum.

As a result, this research will continue by increasing the complexity of the device parameters, improving the automation framework to be able to generate multiple shape devices, and having an intelligent model with multi-objective fitness calculation that can not only predict one numerical representation of a device but that can address the entire inverse design problem, by predicting also the shape of the resonance curve in order to achieve devices with specific spectrum.

REFERENCES

- ANKER, J. N.; HALL, W. P.; LYANDRES, O.; SHAH, N. C.; ZHAO, J.; DUYNE, R. P. V. Biosensing with plasmonic nanosensors. **Nanoscience and Technology: A Collection of Reviews from Nature Journals**, World Scientific, p. 308–319, 2010.
- BRAGA, A. d. P.; LUDERMIR, T. B.; CARVALHO, A. C. P. d. L. F. **Redes neurais artificiais: teoria e aplicações**. [S.l.]: LTC, 2007.
- CELA, C. **The Finite-Difference TimeDomain Method (FDTD)**. 2014. Available at: <<https://my.ece.utah.edu/~ece6340/LECTURES/lecture%2014/FDTD.pdf>>.
- DAHLIN, A. B.; TEGENFELDT, J. O.; HÖÖK, F. Improving the instrumental resolution of sensors based on localized surface plasmon resonance. **Analytical chemistry**, ACS Publications, v. 78, n. 13, p. 4416–4423, 2006.
- FU, P.-H.; LO, S.-C.; TSAI, P.-C.; LEE, K.-L.; WEI, P.-K. Optimization for gold nanostructure-based surface plasmon biosensors using a microgenetic algorithm. **ACS Photonics**, ACS Publications, v. 5, n. 6, p. 2320–2327, 2018.
- HOCHREITER, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. **International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems**, World Scientific, v. 6, n. 02, p. 107–116, 1998.
- HUNTINGTON, M. D.; LAUHON, L. J.; ODOM, T. W. Subwavelength lattice optics by evolutionary design. **Nano letters**, ACS Publications, v. 14, n. 12, p. 7195–7200, 2014.
- LIFESCIENCES, N. **LSPR Technology: The Four Most Frequently Asked Questions**. 2016. Available at: <<https://www.news-medical.net/whitepaper/20161004/LSPR-Technology-The-Four-Most-Frequently-Asked-Questions.aspx#:~:text=What%20is%20LSPR%20and%20how,range%20of%20light%20by%20LSPR.>>>
- LIU, J.; JALALI, M.; MAHSHID, S.; WACHSMANN-HOGIU, S. Are plasmonic optical biosensors ready for use in point-of-need applications? **Analyst**, Royal Society of Chemistry, v. 145, n. 2, p. 364–384, 2020.
- LOPEZ, G. A.; ESTEVEZ, M.-C.; SOLER, M.; LECHUGA, L. M. Recent advances in nanoplasmonic biosensors: applications and lab-on-a-chip integration. **Nanophotonics**, De Gruyter, v. 6, n. 1, p. 123–136, 2017.
- MACHADO, G. L.; TEIXEIRA, F. M.; FERREIRA, G. S.; VERSIANI, A. F.; ANDRADE, L. M.; LADEIRA, L. O.; FONSECA, F. G. da; RAMIREZ, J. C. Computational guided method applied to lspr-based biosensor for specific detection of the four-serotypes of dengue virus in seropositive patients. **Particle & Particle Systems Characterization**, Wiley Online Library, v. 39, n. 3, p. 2100157, 2022.
- MALKIEL, I.; MREJEN, M.; NAGLER, A.; ARIELI, U.; WOLF, L.; SUCHOWSKI, H. Plasmonic nanostructure design and characterization via deep learning. **Light: Science & Applications**, Nature Publishing Group, v. 7, n. 1, p. 1–8, 2018.
- MARKET, B. **Biosensors Market with COVID-19 Impact by Type, Product, Technology, Application and Region - Global Forecast to 2026**. 2021. Available at: <<https://www.marketsandmarkets.com/Market-Reports/biosensors-market-798.html>>.

OH, B.-R.; HUANG, N.-T.; CHEN, W.; SEO, J. H.; CHEN, P.; CORNELL, T. T.; SHANLEY, T. P.; FU, J.; KURABAYASHI, K. Integrated nanoplasmonic sensing for cellular functional immunoanalysis using human blood. **ACS nano**, ACS Publications, v. 8, n. 3, p. 2667–2676, 2014.

SÖNNICHSEN, C.; REINHARD, B. M.; LIPHARDT, J.; ALIVISATOS, A. P. A molecular ruler based on plasmon coupling of single gold and silver nanoparticles. **Nature biotechnology**, Nature Publishing Group, v. 23, n. 6, p. 741–745, 2005.

VERSIANI, A. F.; MARTINS, E. M.; ANDRADE, L. M.; COX, L.; PEREIRA, G. C.; BARBOSA-STANCIOLI, E. F.; NOGUEIRA, M. L.; LADEIRA, L. O.; FONSECA, F. G. da. Nanosensors based on lspr are able to serologically differentiate dengue from zika infections. **Scientific reports**, Nature Publishing Group, v. 10, n. 1, p. 1–17, 2020.

YAO, K.; UNNI, R.; ZHENG, Y. Intelligent nanophotonics: merging photonics and artificial intelligence at the nanoscale. **Nanophotonics**, De Gruyter, v. 8, n. 3, p. 339–366, 2019.