

Benemérita Universidad Autónoma de Puebla  
Facultad de Ciencias Físico Matemáticas

**Implementación de algoritmos de aprendizaje  
automático para la clasificación de osciladores  
bioquímicos acoplados**

Presenta: Antonio Zúñiga Mora  
Asesores: M.C. Jesús Andrés Arzola Flores  
Dr. José Fernando Rojas Rodríguez

Tesis presentada para obtener el grado de: Licenciatura en Física Aplicada

12 de Junio de 2021



**Jurado**

Presidente: Dr. Jorge Velázquez Castro

Secretario: Dr. Adrián Apolinar Hernández Santiago

Vocal: Dra. Esmeralda Vidal Robles

A mis padres.



# Índice general

<b>Resumen</b>	<b>7</b>
<b>Abstract</b>	<b>9</b>
<b>1 Introducción</b>	<b>11</b>
<b>2 Marco teórico</b>	<b>13</b>
2.1 Sistemas de ecuaciones diferenciales ordinarias de primer orden . . . .	13
2.1.1 Teorema de Picard-Lindelöf . . . . .	14
2.2 Métodos numéricos . . . . .	15
2.2.1 Regla de Runge-Kutta . . . . .	15
2.3 Sincronización . . . . .	17
2.3.1 Sincronización maestro-esclavo . . . . .	17
2.4 Descripción de los modelos . . . . .	18
2.4.1 Oregonator . . . . .	19
2.4.2 Brusselator . . . . .	21
2.4.3 Glucólisis y oscilaciones glicolíticas . . . . .	21
2.5 Gráfica de recurrencia . . . . .	24
2.6 Aprendizaje automático (ML) . . . . .	25
2.6.1 Modelos de aprendizaje supervisado . . . . .	27
2.6.2 Técnicas de entrenamiento de aprendizaje supervisado . . . . .	30
2.6.3 Métricas de evaluación . . . . .	32
<b>3 Metodología</b>	<b>37</b>
3.1 Soluciones numéricas . . . . .	38
3.1.1 Soluciones numéricas de los modelos acoplados . . . . .	42
3.2 Gráficas de recurrencia . . . . .	56
3.3 Implementación de los modelos de aprendizaje automático supervisado	58
<b>4 Resultados y discusión</b>	<b>65</b>
4.1 Inception V3 . . . . .	66
4.1.1 Validación cruzada estratificada . . . . .	66
4.1.2 Leave One Out . . . . .	69
4.2 SqueezeNet . . . . .	72
4.2.1 Validación cruzada estratificada . . . . .	72
4.2.2 Leave One Out . . . . .	75

---

<b>5 Conclusiones</b>	<b>79</b>
<b>Agradecimientos</b>	<b>81</b>
<b>Bibliografía</b>	<b>83</b>
<b>Nomenclatura</b>	<b>90</b>

# Índice de figuras

2.1	Perceptrón de una capa oculta. La capa más a la izquierda, conocida como capa de entrada, consta de un conjunto de neuronas $\{x_i x_1, x_2, \dots, x_m\}$ que representa a las entradas o descriptores de la variable a predecir. Cada neurona de la capa oculta transforma los valores de la capa anterior con una suma lineal ponderada $w_1x_1 + w_2x_2 + \dots + w_mx_m$ , seguido de una función de activación no lineal $g(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ . La capa de salida recibe los valores de la última capa oculta y los transforma en valores de salida. . . . .	28
2.2	División de datos en cinco validaciones cruzadas (validación cruzada de 5 pliegues). . . . .	31
2.3	Ejemplo gráfico de una curva ROC característica. El AUC se representa con la parte sombreada. . . . .	34
3.1	Diagrama de flujo que muestra la metodología implementada. Primero se calculan las soluciones numéricas para cada uno de los tres modelos, después, son acoplados entre sí (oregonator con brusselator O-B, oregonator con glucólisis O-G y glucólisis con brusselator G-B) y nuevamente se obtiene la solución numérica del sistema. El resultado de las soluciones numéricas, son en efecto, series temporales para cada una de las variables acopladas ( $x$ y $z$ en O-B, $\sigma_1$ y $x$ en O-G y por último $\sigma_1$ y $z$ en G-B), por lo que, es posible calcular la gráfica de recurrencia para cada una de estas series y por consiguiente, entrenar modelos de aprendizaje automático. . . . .	37
3.2	Solución numérica para el modelo oregonator podemos notar el comportamiento oscilatorio para las dos variables donde la especie $x$ desempeña el papel de activador y la especie $z$ el papel de inhibidor. . .	39
3.3	Solución numérica para el modelo brusselator. Nuevamente se presenta el comportamiento oscilatorio típico de una reacción autocatalítica. En este caso, la especie $z$ desempeña el papel de activador y la especie $y$ el de inhibidor. . . . .	40
3.4	Solución numérica para el modelo de glucólisis mostrando la evolución de $\sigma_1$ y $\sigma_2$ (en el gráfico aparecen como $s_1$ y $s_2$ respectivamente) para el modelo de glucólisis de Sel'kov hacia una solución periódica. . . . .	42
3.5	Solución numérica para el acoplamiento bidireccional lineal de los sistemas oregonator y brusselator para valores de $k = 0.1, 0.2, \dots, 1.1$ .	47

- 3.6 Solución numérica para el acoplamiento bidireccional lineal de los sistemas oregonator con glucólisis y oscilaciones glicolíticas para valores de  $k = 0.1, 0.2, \dots, 1.1$ . Es de notar que a medida que  $k$  aumenta, las oscilaciones comienzan a desaparecer. . . . . 52
- 3.7 Solución numérica para el acoplamiento bidireccional lineal de los sistemas brusselator y glucólisis y oscilaciones glicolíticas para valores de  $k = 0.1, 0.2, \dots, 1.1$ . . . . . 56
- 3.8 Gráficas de recurrencia para  $\sigma_1$  y  $k=0.1$  en el acoplamiento O-G, con umbrales de 0.1, 0.01 y 0.001 (de izquierda a derecha). Notar que a medida que el umbral se hace más pequeño, los puntos negros disminuyen debido a que el umbral es más estricto. En todas las gráficas de recurrencia la diagonal siempre va a aparecer con un punto negro ya que se está comparando el  $i$ -ésimo valor consigo mismo. Para valores cercanos a este, el cambio en las concentraciones es mínimo por lo que es común, encontrar valores cercanos a la diagonal. Lo que identifica a cada acoplamiento de los demás, son los valores lejanos a esta y los patrones que se forman. . . . . 57
- 3.9 Gráficas de recurrencia para  $\sigma_1$ , umbral  $\epsilon = 0.001$  y  $k = 0.1, 0.3, 0.5, 0.7, 0.9$  y 1.1 (de izquierda a derecha y de arriba hacia abajo). Podemos notar que a medida que  $k$  aumenta, se forma un patrón bien definido y representativo del acoplamiento. Para valores de acoplamiento efectivo, se obtienen puntos negros cercanos a la diagonal. Como estamos en el acoplamiento O-G, el amortiguamiento (incisos d) e) y f)) se muestra en un continuo de puntos negros en la sección superior izquierda. . . . 57
- 3.10 Modelo propuesto para la clasificación del acoplamiento bidireccional lineal de sistemas dinámicos. . . . . 59
- 3.11 Interfaces gráficas de cada widget de gestión. La ventana del widget abre un directorio para importar la carpeta que contiene las variables de clase, mientras que el widget de Image Embedding nos permite seleccionar la red CNN para calcular los vectores con los descriptores de imagen. Por último el widget Image Viewer simplemente nos permite navegar entre las gráficas de los diagramas de recurrencia. . . 61
- 3.12 Interfaces gráficas de cada widget de modelos predictivos. En el modelo Logistic Regression el tipo de regularización L2 utiliza la función de verosimilitud (ecuación 2.35). El widget Neural Network nos permite seleccionar el número de neuronas en una capa oculta en el perceptrón multicapa (en este caso 10) con funciones de activación logísticas; el solucionador L-BFGS- B utiliza la ecuación 2.37 y  $\alpha$  es un parámetro asociado a la regularización L2. El widget que aprende un modelo kNN nos permite seleccionar el número de vecinos para realizar la predicción, el tipo de métrica y el peso que tengan cada uno de estos vecinos. Por último, los parámetros del algoritmo SVM, son la penalización  $C = 1/L2$  y  $\xi_i$ , la pérdida en la regresión de la ecuación 2.39, con la función del núcleo nombrada en la ecuación 2.40. 62

3.13 Interfaces gráficas de cada widget de evaluación. El widget Test and Score nos permite seleccionar el tipo de entrenamiento y evaluación (en el caso de la validación cruzada, es posible seleccionar el número de pliegues y si esta será estratificada o no). Por otro lado, el widget Confusion Matrix muestra el número de instancias y cada una de las predicciones de cada modelo. . . . . 63



# Índice de cuadros

2.2	Ejemplo de matriz de confusión para tres clases A, B y C; cada clase cuenta con 53, 55 y 51 datos correspondientemente. Las filas representan la clase real de cada dato a predecir, mostrando el número de datos reales por clase en la última entrada de la fila; mientras que las columnas representan las decisiones tomadas por los modelos de aprendizaje automático, en este caso, la última representa el número de datos por clase según los modelos predictivos. . . . .	33
3.1	Tabla de valores de los parámetros adimensionales en el modelo oregonator. . . . .	38
3.2	Tabla de valores de los parámetros en el modelo brusselator. . . . .	40
3.3	Tabla de valores de los parámetros adimensionales en el modelo de glucólisis. . . . .	41
4.1	Número de diagramas por clase. . . . .	65
4.2	Matrices de confusión para cada modelo predictivo por el método de validación cruzada estratificada de 20 pliegues de la red Inception V3. . . . .	66
4.3	Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase G-B por el método de validación cruzada estratificada de 20 pliegues. . . . .	67
4.4	Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase O-B por el método de validación cruzada de 20 pliegues. . . . .	67
4.5	Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase O-G por el método de validación cruzada estratificada de 20 pliegues. . . . .	68
4.6	Tabla de resultados con el promedio de los resultados AOC en cada clase por el método de validación cruzada estratificada de 20 pliegues. . . . .	68
4.7	Matriz Bayessiana con el análisis entre modelos para el método de validación cruzada de 20 pliegues de la red Inception V3. . . . .	69
4.8	Matrices de confusión para cada modelo predictivo por el método leave one out de la red Inception V3. . . . .	69
4.9	Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase G-B por el método de leave one out. . . . .	70
4.10	Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase O-B por el método de leave one out. . . . .	71
4.11	Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase O-G por el método de leave one out. . . . .	71

4.12	Tabla de resultados con el promedio de los resultados AOC en cada clase por el método de leave one out. . . . .	71
4.13	Matrices de confusión para cada modelo predictivo por el método de validación cruzada de 20 pliegues de la red SqueezeNet. . . . .	72
4.14	Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase G-B por el método de validación cruzada estratificada de 20 pliegues. . . . .	73
4.15	Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase O-B por el método de validación cruzada estratificada de 20 pliegues. . . . .	74
4.16	Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase O-G por el método de validación cruzada estratificada de 20 pliegues. . . . .	74
4.17	Tabla de resultados con el promedio de los resultados AOC en cada clase por el método de validación cruzada estratificada de 20 pliegues. . . . .	74
4.18	Matriz Bayessiana con el análisis entre modelos para el método de validación cruzada estratificada de 20 pliegues de la red SqueezeNet. . . . .	75
4.19	Matrices de confusión para cada modelo predictivo por el método de validación cruzada de 20 pliegues de la red SqueezeNet. . . . .	75
4.20	Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase G-B por el método de leave one out. . . . .	76
4.21	Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase O-B por el método de leave one out. . . . .	77
4.22	Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase O-G por el método de leave one out. . . . .	77
4.23	Tabla de resultados con el promedio de los resultados AOC en cada clase por el método leave one out. . . . .	77

# Resumen

**Palabras clave:** *sincronización, oregonator, brusselator, aprendizaje automático supervisado, glucólisis, acoplamiento.*

La identificación del acoplamiento de un grupo de procesos fisiológicos es de vital importancia en la medicina moderna, dado que el desequilibrio de ellos es el resultado de la aparición de enfermedades crónico degenerativas. Los procesos fisiológicos generalmente son regulados por una intrincada red de reacciones bioquímicas, por lo que el estudio de su acoplamiento es demasiado complicado.

Los avances en los métodos de aprendizaje supervisado, permiten crear modelos capaces de aprender a clasificar patrones a partir de datos obtenidos de series temporales, y de esta manera, lograr la predicción de estados de acoplamiento entre procesos fisiológicos con alta precisión. El modelo de aprendizaje supervisado requiere únicamente de un conjunto de datos previamente clasificados que le permita al algoritmo realizar la clasificación. Una manera de obtener un conjunto de datos confiable para el entrenamiento de los algoritmos de aprendizaje supervisado es a través del cálculo de los diagramas de recurrencia, los cuales pueden ser obtenidos de las series de tiempo resultado de la solución numérica de sistemas dinámicos asociados al acoplamiento de distintos procesos fisiológicos.

En este trabajo de investigación se estudia el acoplamiento de tres modelos matemáticos relacionados con procesos fisiológicos: glucólisis y oscilaciones glicolíticas, brusselator y oregonator; mediante un acoplamiento bidireccional lineal para distintos valores del parámetro de acoplamiento. Posteriormente, se calcula la gráfica de recurrencia a partir de la solución numérica de los sistemas acoplados. Dichas gráficas de recurrencia se emplearon para entrenar cuatro modelos de aprendizaje supervisado: k-vecinos cercanos (KNN por sus siglas en inglés), máquina de soporte vectorial (SVM, por sus siglas en inglés), red neuronal perceptrón multicapa y regresión logística, con la finalidad de clasificar los modelos dinámicos acoplados; los modelos son entrenados y examinados por los métodos de validación cruzada estratificada y leave one out; empleando las redes neuronales convolucionales (CNN, por sus siglas en inglés) Inception V3 y SqueezeNet para definir los descriptores de cada gráfica de recurrencia. Los resultados son evaluados en cinco métricas (AUC, CA, F1, precisión y recall).

Esta metodología nos ha permitido obtener porcentajes de exactitud cercanos al 100 % en la clasificación de sistemas dinámicos acoplados. Además, puede extenderse al campo experimental para la clasificación de estados de acoplamiento en osciladores fisiológicos y de esta manera asociar patologías a dichos estados.



# Abstract

**Keywords:** *synchronization, oregonator, brusselator, supervised machine learning, glycolysis, coupling.*

The identification of the coupling of a group of physiological processes is of vital importance in modern medicine, since their imbalance is the result of the appearance of chronic degenerative diseases. Physiological processes are generally regulated by an intricate network of biochemical reactions, making the study of their coupling too complicated.

Advances in supervised learning methods makes possible the creation of models capable of learning to classify patterns from data obtained from time series, and in this way, achieve the prediction of coupling states between physiological processes with high precision. The supervised learning model requires only a previously classified data set that allows the algorithm to perform the classification. One way to obtain a reliable data set for the training of supervised learning algorithms is through the calculation of recurrence diagrams, which can be obtained from the time series resulting from the numerical solution of dynamic systems associated with the coupling of different physiological processes.

In this research work the coupling of three mathematical models (glycolysis and glycolytic oscillations, brusselator and oregonator) related to physiological processes, is studied; using a bidirectional linear coupling for different values of the coupling parameter. Subsequently, the recurrence graph is calculated from the numerical solution of the coupled systems. These recurrence plots were used to train four supervised learning models: k-nearest neighbors (KNN), support vector machine (SVM), multilayer perceptron neural network and logistic regression; the models are trained and examined by stratified cross-validation and leave one out methods, using convolutional neural networks (CNN) Inception V3 and SqueezeNet to define the descriptors for each recurrence plot. The results are evaluated in five metrics (AUC, CA, F1, precision and recall).

This methodology has allowed us to obtain percentages of accuracy close to 100% in the classification of coupled dynamic systems. In addition, the methodology can be extended to the experimental field for the classification of coupling states in physiological oscillators and in this way associate pathologies with those states.



# 1 Introducción

La aparición de enfermedades crónico degenerativas (como lo son el cáncer, la resistencia a la insulina, diabetes mellitus II, entre otras) es consecuencia del desajuste de una amplia variedad de procesos fisiológicos que a su vez están regulados por una compleja red de reacciones químicas [6], lo que dificulta el estudio de la interacción, el acoplamiento y la activación que pueden existir entre diferentes procesos fisiológicos; por lo cual, es necesario desarrollar herramientas que permitan identificar cuándo los procesos fisiológicos se encuentran acoplados o desacoplados [6, 13, 37, 48].

La teoría de sistemas dinámicos juega un papel relevante en la física actual, y por supuesto, en los sistemas fisiológicos [4]. Los sistemas dinámicos pueden definirse como una estructura cuyas variables de estado siguen una serie de reglas temporales. Todos los sistemas dinámicos pueden ser descritos por un conjunto de ecuaciones diferenciales y la solución de dichas ecuaciones representa la evolución del sistema en cualquier momento [42]. Por otro lado, los complejos mecanismos bioquímicos relacionados con distintos procesos fisiológicos pueden ser vistos como sistemas dinámicos, por lo que estudiar los procesos mediante los cuales estos pueden sincronizarse es de vital importancia para el campo de la biomedicina.

El fenómeno de sincronización de sistemas dinámicos ha recibido mucha atención por parte de la comunidad científica en los últimos veinte años, debido a que al parecer es una propiedad emergente en muchos sistemas, tales como los distintos procesos fisiológicos. Hoy en día, existen muchas herramientas que permiten identificar estados de sincronización en sistemas dinámicos [7] [27] [45] [14], tales como el cálculo de la dimensión fractal de las series de tiempo [17], los exponentes de Lyapunov [17], la autocorrelación [44], etc. Sin embargo, actualmente las herramientas de la inteligencia artificial comienzan a tomar mayor importancia en el estudio de fenómenos fisiológicos [14] [29].

Actualmente, gracias al desarrollo de la inteligencia artificial, especialmente, del aprendizaje automático, es posible dotar de cierta capacidad de aprendizaje a algoritmos computacionales con la finalidad de reconocer y predecir una gran cantidad de patrones. Por lo que, en el presente trabajo se propone la implementación de

distintos algoritmos de aprendizaje automático para la predicción del acoplamiento entre distintos sistemas dinámicos, a partir del uso de los diagramas de recurrencia producto de las series de tiempo de la solución numérica de dichos sistemas.

## 2 Marco teórico

En el marco teórico de este trabajo, se presenta una breve introducción sobre la teoría de sistemas dinámicos. Después, se discute el proceso de sincronización entre sistemas dinámicos oscilante. A continuación, se detallan los tres modelos dinámicos que serán utilizados para entrenar al algoritmo de aprendizaje supervisado; también se expone la herramienta de la gráfica de recurrencia y su importancia en la dinámica oscilatoria. Por último, se explica el proceso de aprendizaje automático y sus tipos, al mismo tiempo que se enlistan los modelos, las técnicas de entrenamiento y las métricas de evaluación, de aprendizaje supervisado.

### 2.1. Sistemas de ecuaciones diferenciales ordinarias de primer orden

La mayoría de los sistemas que nos rodean están sometidos al cambio en relación al tiempo, las ecuaciones diferenciales son una herramienta matemática que nos permiten sintetizar en una ecuación la dinámica de las variables de un sistema y conocer su evolución. A dichas variables del sistema se les conoce como *variables de estado* y son el objeto de interés del estudio de sistemas que evolucionan con el tiempo [15].

Para este trabajo, estamos interesados en *sistemas de ecuaciones diferenciales ordinarias de primer orden*. Un sistema de ecuaciones diferenciales ordinarias es un conjunto de ecuaciones diferenciales compuesto de una serie de condiciones iniciales; en cambio, un conjunto solución del sistema de ecuaciones, es un conjunto de funciones que satisfacen todas y cada una de las ecuaciones diferenciales del sistema y cada una de las condiciones iniciales.

La forma general de un sistema de  $n$  ecuaciones diferenciales ordinarias de primer

orden con  $n$  variables de estado es la siguiente:

$$\begin{cases} \frac{dx_1}{dt} = F_1(x_1, x_2, \dots, x_n; t) \\ \frac{dx_2}{dt} = F_2(x_1, x_2, \dots, x_n; t) \\ \dots \\ \frac{dx_n}{dt} = F_n(x_1, x_2, \dots, x_n; t) \end{cases} \quad (2.1)$$

Con las condiciones iniciales:

$$x_1(t_0) = x_{1,0}, \quad x_2(t_0) = x_{2,0}, \quad \dots, \quad x_n(t_0) = x_{n,0} \quad (2.2)$$

### 2.1.1. Teorema de Picard-Lindelöf

Una manera de asegurarnos que los sistemas de ecuaciones diferenciales ordinarias de primer orden tienen una solución única, es a través del teorema de Picard-Lindelöf.

Consideremos el problema de Cauchy, también conocido como *problema del valor inicial*, para una ecuación diferencial definida sobre  $\mathbb{R}^n$ :

$$\begin{cases} \frac{dx}{dt} = f(t, x) \\ x(t_0) = x_0 \end{cases} \quad (2.3)$$

Donde  $f(t, x) : \Omega \subseteq \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  es una función continua y localmente Lipschitz respecto de  $x$ , y  $\Omega$  es abierto.

Entonces, el teorema de Picard-Lindelöf asegura que, dado  $(t_0, x_0) \in \Omega$  podemos encontrar un intervalo cerrado  $I_\alpha = [t_0 - \alpha, t_0 + \alpha] \subset \mathbb{R}$ ,  $\alpha > 0$ , donde existe solución única del problema de Cauchy; que a la vez cumple que los pares  $(t, x(t)) \in \Omega$ ,  $\forall t \in I_\alpha$ .

Notar que el teorema de Picard puede ser generalizado para un sistema de  $n$  ecua-

ciones diferenciales con  $n$  incógnitas [36].

## 2.2. Métodos numéricos

Existen ecuaciones diferenciales lineales cuya solución analítica es difícil de obtener, por lo que, para encontrar dicha solución, es necesaria la implementación de métodos numéricos.

Consideremos nuevamente el problema del valor inicial:

$$\frac{dy}{dt} = f(t, y(t)), \quad a \leq t \leq b, \quad y(a) = y_0 \quad (2.4)$$

buscamos una función  $y(t) \in C^1 [a, b]$  que satisfaga la identidad anterior. Asumiendo que  $f(t, y(t))$  está definida para  $t \in [a, b]$  y  $y \in \mathbb{R}^m$ . Podemos encontrar dicha función  $y(t)$  basándonos en el teorema siguiente:

Si  $f(t, y)$  es continua en  $t \in [a, b]$  y  $y \in \mathbb{R}$  satisface la condición de Lipschitz:

$$\|f(t, y) - f(t, y^*)\| \leq L \|y - y^*\| \quad (2.5)$$

donde  $t \in [a, b]$ , y  $y, y^* \in R$  entonces la ecuación (2.4) tiene una única solución:  $y(t)$ , donde  $a \leq t \leq b$  para cualquier  $y_0 \in R$ .

Así, suponiendo el problema (2.4), podemos considerar una aproximación  $(t_n, y(t_n))$  por pasos de la siguiente manera:

$$y_{n+1} = y_n + h \cdot \Phi(t_n, y_n; h), \quad h > 0. \quad (2.6)$$

donde la función  $\Phi$  se puede ver como el incremento aproximado en cada paso y define los métodos numéricos denominados paso a paso [2].

### 2.2.1. Regla de Runge-Kutta

El método de Runge-Kutta es un método paso a paso diseñado para sustituir las expansiones de Taylor pero usando solo evaluaciones de la función  $f(t, y)$  y no en

sus derivadas, en la resolución del problema de valor inicial. Es un algoritmo que nos permite resolver numéricamente el problema:  $y' = f(t, y)$  con  $y(t_0) = y_0$  a través de un promedio ponderado de valores de  $f(t, y)$  tomados en diferentes puntos del intervalo  $t_n \leq t \leq t_{n+1}$ . El método de Runge-Kutta de orden 4 se representa de la siguiente manera:

$$\begin{cases} y_0 = y(a) \\ y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad n = 0, 1, 2, \dots \end{cases} \quad (2.7)$$

siendo:

$$\Phi(x_n, y_n) = k_1 + 2k_2 + 2k_3 + k_4; \quad h = \frac{1}{6} \quad (2.8)$$

la función incremento para la solución en el problema del valor inicial. Donde:

$$\begin{aligned} k_1 &= hf(t_n, y_n) \\ k_2 &= hf\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right) \\ k_3 &= hf\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right) \\ k_4 &= hf(t_n + h, y_n + k_3) \end{aligned} \quad (2.9)$$

Los errores local y global son en este caso proporcionales a  $h^5$  y  $h^4$  respectivamente.

El método de Runge-Kutta planteado puede ser generalizado para sistemas de ecuaciones de orden superior al primero [2].

Consideremos el siguiente sistema de dos ecuaciones de primer orden:

$$\begin{cases} \frac{dx}{dt} = f(t, x, y) \\ \frac{dy}{dt} = g(t, x, y) \\ x(0) = x_0 \\ y(0) = y_0 \end{cases} \quad (2.10)$$

La extensión natural del método de Runge-Kutta nos conduce a las siguientes expresiones:

$$\begin{aligned}x_{n+1} &= x_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ y_{n+1} &= y_n + \frac{1}{6}(l_1 + 2l_2 + 2l_3 + l_4)\end{aligned}\tag{2.11}$$

donde:

$$\begin{aligned}k_1 &= hf(t_0, x_n, y_n); \\ k_2 &= hf(t_0 + \frac{h}{2}, x_n + \frac{h}{2}, y_n + \frac{k_1}{2}); \\ k_3 &= hf(t_0 + \frac{h}{2}, x_n + \frac{h}{2}, y_n + \frac{k_2}{2}); \\ k_4 &= hf(t_0 + h, x_n + h, y_n + k_3);\end{aligned}\tag{2.12}$$

$$\begin{aligned}l_1 &= hg(t_0, x_n, y_n) \\ l_2 &= hg(t_0 + \frac{h}{2}, x_n + \frac{h}{2}, y_n + \frac{k_1}{2}) \\ l_3 &= hg(t_0 + \frac{h}{2}, x_n + \frac{h}{2}, y_n + \frac{k_2}{2}) \\ l_4 &= hg(t_0 + h, x_n + h, y_n + k_3)\end{aligned}\tag{2.13}$$

Notar que el método puede generalizarse para sistemas de  $n$  funciones con  $n$  variables de estado.

## 2.3. Sincronización

El acoplamiento de los procesos fisiológicos, es un ejemplo de un fenómeno más general conocido como de sincronización.

La sincronización representa el proceso de interacción entre dos o más osciladores en el que se lleva a cabo el ajuste de sus ritmos de oscilación, resultando en una evolución conjunta de algunas de sus propiedades dinámicas. Los fenómenos de sincronización ocurren principalmente entre osciladores autosostenidos, a través de algún tipo de acoplamiento entre ellos [26].

Desde la sincronización de nuestras funciones fisiológicas con el ciclo día y noche, hasta el ajuste de miles de células marcapasos en el nodo sinoauricular, para disparar al unísono y regular los latidos de nuestro corazón, los fenómenos de sincronización son parte común de nuestra vida diaria.

### 2.3.1. Sincronización maestro-esclavo

El método de sincronización maestro-esclavo o acoplamiento conductivo, consiste en lo siguiente: en lugar de pensar en un solo sistema en evolución de dos condiciones

iniciales diferentes, considere dos copias idénticas del sistema (junto con sus respectivas condiciones iniciales), y relacionarlas mediante alguna *función de acoplamiento* [45].

Supongamos que trabajamos con un sistema con tres variables de estado, de esta forma el sistema está definido en  $\mathbb{R}^3$ , Sea  $\mathbf{x} = (x, y, z)$  las coordenadas generalizadas correspondientes al primer sistema que será elegido como el *maestro*, y  $\mathbf{x}_s = (x_s, y_s, z_s)$  los del segundo, el sistema *esclavo*. Y sus evoluciones determinadas por el mismo campo vectorial  $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ ; entonces tenemos que:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)), \quad \dot{\mathbf{x}}_s(t) = \mathbf{f}(\mathbf{x}_s(t)) \quad (2.14)$$

Las posibilidades de definir la función de acoplamiento son muchas, pero una muy popular, consiste en tomar una linealización en la diferencia  $x - x_s$ , con coeficientes posiblemente dependientes de las variables maestras, de esta manera [7]:

$$\mathbf{k}(\mathbf{x}) \cdot (\mathbf{x} - \mathbf{x}_s) = \begin{pmatrix} k_{11}(\mathbf{x}) & k_{12}(\mathbf{x}) & k_{13}(\mathbf{x}) \\ k_{21}(\mathbf{x}) & k_{22}(\mathbf{x}) & k_{23}(\mathbf{x}) \\ k_{31}(\mathbf{x}) & k_{32}(\mathbf{x}) & k_{33}(\mathbf{x}) \end{pmatrix} \cdot \begin{pmatrix} x - x_s \\ y - y_s \\ z - z_s \end{pmatrix} \quad (2.15)$$

Esta función de acoplamiento debe ser restada al subsistema esclavo para que responda a la acción del subsistema maestro y sumada al subsistema maestro para regir la dinámica del sistema entero (*acoplamiento bidireccional*). Por lo tanto, el sistema completo puede ser descrito de la siguiente manera:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{k}(\mathbf{x}) \cdot (\mathbf{x} - \mathbf{x}_s) \\ \dot{\mathbf{x}}_s(t) = \mathbf{f}(\mathbf{x}_s(t)) - \mathbf{k}(\mathbf{x}) \cdot (\mathbf{x} - \mathbf{x}_s) \end{cases} \quad (2.16)$$

## 2.4. Descripción de los modelos

Sabiendo que una manera de sincronizar dos sistemas de ecuaciones diferenciales, es a través de una función de acoplamiento, podemos pensar en la modelación de un par de sistemas que representen procesos fisiológicos para después sincronizarlos a través de dicha función.

Tres modelos importantes para la física, biología y química, que representan procesos fisiológicos, son discutidos en este apartado y representados a través de sistemas de ecuaciones diferenciales ordinarias de primer orden.

### 2.4.1. Oregonator

El modelo conocido como Oregonator es la manera más simple y realista de modelar la dinámica oscilatoria de la *reacción de Belousov-Zhabotinsky* (reacción B-Z) y ha sido utilizado para estudiar los mecanismos de retroalimentación bioquímica presentes en el ciclo de Krebs, de ahí su importancia [41].

La reacción B-Z es una familia de reacciones químicas oscilantes. Durante estas reacciones, los iones de metales de transición catalizan la oxidación de varios reductores, generalmente orgánicos, por el ácido brómico en una solución ácida. La mayoría de las reacciones BZ son homogéneas y permiten observar el desarrollo de patrones complejos en el tiempo y el espacio a simple vista, debido a que su escala espacial es correspondiente a varios milímetros y su escala temporal correspondiente a una docena de segundos. También, la reacción es capaz de generar hasta varios miles de ciclos oscilatorios en un sistema cerrado, lo que permite estudiar ondas y patrones químicos sin una reposición constante de reactivos [24].

Belousov descubrió la primera reacción BZ con el par  $\text{Ce}^{3+} / \text{Ce}^{4+}$  como catalizador y ácido cítrico como reductor. Observó que el color de la solución de reacción oscilaba entre incoloro y amarillo y encontró que la frecuencia de las oscilaciones aumentaba con el aumento de temperatura. Zhabotinsky reemplazó el ácido cítrico con ácido malónico (MA) para crear la versión más popular de la reacción BZ. En general, Belousov sugirió que la reacción puede ser descrita por dos partes principales: la oxidación autocatalítica de  $\text{Ce}^{3+}$  por  $\text{HBrO}_3$  y la reducción de  $\text{Ce}^{4+}$  por MA y sus derivados del bromo, que son producidos durante la reacción general. En su esquema, la reducción de  $\text{Ce}^{4+}$  va acompañada de la producción de  $\text{Br}^-$  a partir de los derivados del bromo de MA.  $\text{Br}^-$  es un fuerte inhibidor de la oxidación autocatalítica de  $\text{Ce}^{3+}$  debido a su rápida reacción con el autocatalizador,  $\text{HBrO}_2$  [12].

Un ciclo oscilatorio de la reacción BZ se puede describir cualitativamente de la siguiente manera: sea un sistema en el cual, se encuentre una concentración suficientemente alta de  $\text{Ce}^{4+}$ , entonces  $\text{Br}^-$  será producido rápidamente y su concentración también será alta. Como resultado, la oxidación autocatalítica de  $\text{Ce}^{3+}$  se inhibe completamente y la  $[\text{Ce}^{4+}]$  disminuye debido a su reducción  $[\text{Ce}^{4+}]$ . Cuando  $[\text{Ce}^{4+}]$  alcanza su umbral más bajo, la concentración de iones de bromuro cae abruptamente. La oxidación autocatalítica rápida comienza y ahora  $[\text{Ce}^{4+}]$  aumenta. Cuando  $[\text{Ce}^{4+}]$  alcanza su umbral más alto  $[\text{Br}^-]$  aumenta bruscamente e inhibe la oxidación autocatalítica de  $\text{Ce}^{3+}$ . Después, el ciclo se repite [48] [19].

El modelo estándar de Oregonator se escribe comúnmente en términos de cinco pasos

irreversibles [32]:



En este esquema,  $A=\text{BrO}_3^-$ ,  $B$ =especies orgánicas tales como ácido malónico y ácido bromomalónico,  $P = \text{HOBr}$ ,  $X=\text{HBrO}_2$ ,  $Y=\text{Br}^-$ ,  $Z$ =forma oxidada del catalizador y  $f$  es un factor estequiométrico que sirve como parámetro ajustable. Las ecuaciones de velocidad de reacción correspondientes son:

$$\begin{aligned}
 \frac{d[X]}{dt} &= k_1[A][Y] - k_2[X][Y] + k_3[A][X] - 2k_4[X]^2 \\
 \frac{d[Y]}{dt} &= -k_1[A][Y] - k_2[X][Y] + \frac{1}{2}fk_c[B][Z] \\
 \frac{d[Z]}{dt} &= 2k_3[A][X] - k_c[B][Z]
 \end{aligned} \tag{2.18}$$

Ahora, considerando los factores de conversión adimensional de la siguiente manera:  $x = \frac{2k_4[X]}{k_5[A]}$ ,  $y = \frac{k_2[Y]}{k_5[A]}$ ,  $z = \frac{k_c k_4[B][X]}{(k_5[A])^2}$  y  $t = k_c[B]T$ ; las ecuaciones 2.18 se transforman en :

$$\begin{aligned}
 \frac{dx}{dt} &= qy - xy + x(1-x) \\
 \frac{dy}{dt} &= \frac{-qy - xy + fz}{\varepsilon} \\
 \frac{dz}{dt} &= x - z
 \end{aligned} \tag{2.19}$$

Donde  $\varepsilon = \frac{k_c[B]}{k_5[A]}$ ,  $\varepsilon' = \frac{2k_4 k_c[B]}{k_2 k_5[A]}$ ,  $q = \frac{2k_3 k_4}{k_2 k_5}$ . Ahora debido a que  $\varepsilon'$  es muy pequeño comparado con  $\varepsilon$  y las escalas temporales que gobiernan la evolución de  $X = \text{HBrO}_2$  y  $Y = \text{Br}^-$ , podemos utilizar la aproximación de estado estacionario para la variable  $y$ , obteniendo:  $y = fzq + x$ , y al sustituir su valor en el sistema 2.19, obtenemos:

$$\begin{aligned}
 \varepsilon \frac{dx}{dt} &= x(1-x) + f \frac{q-x}{q+x} z \\
 \frac{dz}{dt} &= x - z
 \end{aligned} \tag{2.20}$$

Escrito de esta manera, al reducir el modelo Oregonator, podemos observar que tiene la forma típica de un modelo *activador-inhibidor*, con  $x$  la especie autocatalítica y

$z$  la especie consumidora (inhibidor).

### 2.4.2. Brusselator

El modelo conocido como Brusselator o trimolecular es un modelo hipotético de una reacción química oscilante autocatalítica. Es utilizado debido a su simplicidad teórica, ya que permite conservar la forma funcional de redes más complejas.

Este modelo fue propuesto por Ilya Prigogine y sus colaboradores en la Universidad Libre de Bruselas (Université Libre de Bruxelles), su nombre es un acrónimo de Bruselas y oscilador [21]. El mecanismo de reacción del Brusselator describe un sistema químico hipotético que convierte una especie química A en otra especie D en cuatro pasos, mediante la participación de cuatro especies intermedias, Z, Y, B y C.

El mecanismo está representando por el siguiente conjunto de reacciones:



Las dos especies de interés son Z y Y. Las ecuaciones diferenciales correspondientes a este conjunto de reacciones son:

$$\begin{aligned} \frac{d[Z]}{dt} &= k_1 [A] - k_2 [B] [Z] + k_3 [Y] [Z]^2 - k_4 [Z] \\ \frac{d[Y]}{dt} &= -k_2 [B] [Z] - k_3 [Y] [Z]^2 \end{aligned} \tag{2.22}$$

donde  $k_i$ ,  $i = 1, 2, 3$ , son las constantes de velocidad de cada reacción en 2.21. Para este análisis, se supone que todas las constantes de velocidad son iguales a 1, y se supone que los reactivos A y B tienen un gran exceso de modo que sus concentraciones no cambian con el tiempo. Así, en forma adimensional, el sistema de ecuaciones 2.22 se transforma en :

$$\begin{aligned} \frac{dz}{dt} &= 1 - (b + 1)z + ayz^2 \\ \frac{dy}{dt} &= bz - ayz^2 \end{aligned} \tag{2.23}$$

### 2.4.3. Glucólisis y oscilaciones glicolíticas

El metabolismo es el proceso por el cual se extrae energía útil de los enlaces químicos.

Una ruta metabólica es la sucesión de reacciones enzimáticas necesarias para poder transferir energía química de una forma a otra, en otras palabras, es la secuencia de reacciones químicas donde un sustrato inicial se transforma y da lugar a productos finales[3].

El portador más común de energía en la célula, es el nucleótido adenosín trifosfato (ATP). El ATP se forma por fosforilación de dos maneras diferentes, por la adición de un grupo fosfato inorgánico ( $\text{HPO}_4^{2-}$ ) al difosfato de adenosina (ADP), o mediante la adición de dos grupos fosfato inorgánicos al monofosfato de adenosina (AMP). Dado que los tres grupos fosfato en el ATP tienen cargas negativas, se requiere una energía adicional para superar la repulsión natural de los fosfatos también cargados negativamente a medida que son agregados grupos adicionales a AMP. Por lo tanto, la hidrólisis de ATP a ADP libera grandes cantidades de energía.

Por otro lado, la oxidación de la glucosa está asociada con una gran energía libre negativa,  $\Delta G^0 = -2878.41 \text{ kJ/mol}$ , parte del cual se disipa en forma de calor, sin embargo, en las células vivas gran parte de esta energía libre se almacena en ATP, con una molécula de glucosa que da como resultado 38 moléculas de ATP. El proceso de oxidación de la glucosa es relevante dado que es el encargado de posibilitarle la energía para realizar cualquier trabajo químico a la célula. El proceso de oxidación de la glucosa consiste en tres etapas principales: *glucólisis*, *el ciclo de Krebs* y *la cadena de transporte de electrones*.

La glucólisis consiste en once reacciones elementales, siendo cada una de ellas, una reacción enzimática. Para nuestro modelo, nos enfocamos en los tres primeros pasos [23]:

1. La fosforilación de glucosa a glucosa 6-fosfato.
2. La isomerización de glucosa 6-fosfato a fructosa 6-fosfato.
3. La fosforilación de fructosa 6-fosfato a fructosa 1, 6-bisfosfato.

Esta tercera reacción es catalizada por la enzima fosfofructoquinasa (PFK1). El ATP es tanto un sustrato de PFK1, que se une a un sitio catalítico, como un inhibidor alostérico, que se une a un sitio regulador. La inhibición debida a ATP es eliminada por AMP y, por lo tanto, la actividad de PFK1 aumenta a medida que disminuye la proporción de ATP a AMP. Esta retroalimentación permite que PFK1 regule la tasa de glucólisis en función de la disponibilidad de ATP. Si los niveles de ATP caen, la actividad de PFK1 aumenta aumentando así la tasa de producción de ATP, mientras que, si los niveles de ATP se elevan, la actividad de PFK1 cae y detiene la producción de ATP.

A medida que PFK1 fosforila la fructosa 6-fosfato, el ATP se convierte en ADP;

ADP, a su vez, es convertido de nuevo a ATP y AMP por la reacción:



catalizada por la enzima adenilato quinasa. Dado que normalmente hay poco AMP en las células, la conversión de ADP en ATP y AMP sirve para disminuir significativamente la Relación ATP / AMP, activando así PFK1. Este es un ejemplo de un ciclo de retroalimentación positiva; cuanto mayor es la actividad de PFK1, menor es la relación ATP / AMP, lo que aumenta aún más la actividad de PFK1.

En 1968, Sel'kov propuso un modelo matemático que describe las oscilaciones, que posteriormente fue modificado por Goldbeter y Lefever en 1972. El modelo captura la retroalimentación positiva de ADP debido a la presencia de PFK1. En el modelo, PFK1 está inactivo en su estado libre, pero es activado al unirse con las moléculas de ADP. Por simplicidad, la conversión de ADP a AMP y ADP a ATP no es tomada en cuenta, sin embargo, se asume que la activación de PFK1 por ADP es directa, de esta manera, el efecto en general es similar. Es posible modelar el proceso con las siguientes reacciones [23]:



donde PFK1 (representado por E) es activado o desactivado al unirse con  $\gamma$ -moléculas de ADP (denotado por  $S_2$ ).

Por otro lado, se supone que existe un suministro constante de ATP ( $S_1$ ) mientras que la producto de  $S_2$  se elimina de forma irreversible, y  $S_1$  puede unirse con la forma activa de la enzima para producir como resultado una molécula de  $S_2$ , así:



Aplicando la ley de acción de masas, somos capaces de construir cuatro ecuaciones diferenciales (podemos notar que no necesitamos una quinta ecuación dado que la concentración total de la enzima  $e_0$  se conserva:  $e + x_1 + x_2 = e_0$ ) para la producción de cuatro de las cinco especies:

$$\begin{aligned} \frac{ds_1}{dt} &= v_1 - k_1 s_1 x_1 + k_{-1} x_2 \\ \frac{ds_2}{dt} &= k_2 x_2 - \gamma k_3 s_2^\gamma e + \gamma k_{-3} x_2 - v_2 s_2 \\ \frac{dx_1}{dt} &= -k_1 s_1 x_1 + (k_{-1} + k_2) x_2 + k_3 s_2^\gamma e - k_{-3} x_1 \\ \frac{dx_2}{dt} &= k_1 s_1 x_1 - (k_{-1} + k_2) x_2 \end{aligned} \quad (2.27)$$

donde  $x_1 = [ES_2^\gamma]$ ,  $x_2 = [S_1ES_2^\gamma]$ ,  $s_1 = [S_1]$ ,  $s_2 = [S_2]$  y  $e = [E]$ .

Ahora introduciendo las siguientes variables adimensionales:  $\sigma_1 = \frac{k_1 s_1}{k_2 + k_{-1}}$ ,  $\sigma_2 = (\frac{k_3}{k_{-3}})^{1/\gamma} s_2$ ,  $u_1 = \frac{x_1}{e_0}$ ,  $u_2 = \frac{x_2}{e_0}$ ,  $t = \frac{k_2 + k_{-1}}{e_0 k_1 k_2} \tau$ , encontramos que el sistema de ecuaciones diferenciales (2.27) se transforma en:

$$\begin{aligned} \frac{d\sigma_1}{d\tau} &= v - \frac{k_2 + k_{-1}}{k_2} u_1 \sigma_1 + \frac{k_{-1}}{k_2} u_2 \\ \frac{d\sigma_2}{d\tau} &= \alpha \left[ u_2 - \frac{\gamma k_{-3}}{k_2} \sigma_2^\gamma (1 - u_1 - u_2) + \frac{k_{-3}}{k_2} u_1 \right] - \eta \sigma_2 \\ \epsilon \frac{du_1}{d\tau} &= u_2 - \sigma_1 u_1 + \frac{k_{-3}}{k_2 + k_{-1}} [\sigma_2^\gamma (1 - u_1 - u_2) - u_1] \\ \epsilon \frac{du_2}{d\tau} &= \sigma_1 u_1 - u_2 \end{aligned} \quad (2.28)$$

donde  $\epsilon = \frac{e_0 k_1 k_2}{(k_2 + k_{-1})^2}$ ,  $v = \frac{v_1}{k_2 e_0}$ ,  $\eta = \frac{v_2 (k_2 + k_{-1})}{k_1 k_2 e_0}$ ,  $\alpha = \frac{k_2 + k_{-1}}{k_1} (\frac{k_3}{k_{-3}})^{1/\gamma}$ . Ahora para simplificar el sistema podemos suponer que  $\epsilon$  es muy pequeño, así, es posible despejar a  $u_1$  y  $u_2$  y encontrar sus valores cuasiestables, de la siguiente manera:

$$\begin{aligned} u_1 &= \frac{\sigma_2^\gamma}{\sigma_2^\gamma \sigma_1 + \sigma_2^\gamma + 1} \\ u_2 &= \frac{\sigma_1 \sigma_2^\gamma}{\sigma_2^\gamma \sigma_1 + \sigma_2^\gamma + 1} = f(\sigma_1, \sigma_2) \end{aligned} \quad (2.29)$$

y con estos valores cuasi-estables, la evolución de  $\sigma_1$  y  $\sigma_2$  se simplifica con el siguiente sistema de ecuaciones diferenciales:

$$\begin{aligned} \frac{d\sigma_1}{d\tau} &= v - f(\sigma_1, \sigma_2) \\ \frac{d\sigma_2}{d\tau} &= \alpha f(\sigma_1, \sigma_2) - \eta \sigma_2 \end{aligned} \quad (2.30)$$

## 2.5. Gráfica de recurrencia

La gráfica de recurrencia es una herramienta de análisis que permite revelar la existencia de patrones en series temporales [11]. Para obtener la gráfica de recurrencia de un sistema es necesario conocer el espacio fase correspondiente a dicho sistema, además de calcular la norma entre dos vectores pertenecientes a dicho espacio. Suponemos que  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  es un vector en  $R^n$ . La norma Euclidiana se define como:

$$\|\mathbf{x}\| = \sum_{i=1}^n |x_i| \quad (2.31)$$

Consideremos una trayectoria en el espacio fase descrita por el siguiente conjunto de vectores:  $A = \{\mathbf{y}_i\}_{i=1}^N$ , donde el índice  $i$  está asociado con el tiempo en la gráfica del espacio fase en que ocurre determinado vector. Para calcular la gráfica de recurrencia de dicha trayectoria, es necesario construir previamente la denominada *matriz de recurrencia* definida de la siguiente manera:

$$R_{ij}(\epsilon) = \theta(\epsilon - \|\mathbf{y}_i - \mathbf{y}_j\|) \quad (2.32)$$

para  $i, j = 1, 2, \dots, N$ , donde  $N$  es el total de vectores en el espacio fase,  $\epsilon$  es una distancia umbral y  $\theta$  es la función de Heaviside:

$$\theta(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.33)$$

en general, la matriz de recurrencia  $R_{ij}(\epsilon)$  compara estados en el espacio fase. Si los estados de un sistema son similares o cercanos en el espacio fase (la distancia entre vectores es menor a  $\epsilon$ ), la entrada se marca con un 1. Si los estados son diferentes o están alejados (la distancia entre vectores es mayor a  $\epsilon$ ) la entrada se indica con un 0.

La gráfica de recurrencia corresponde a una gráfica en  $R^2$ , y se construye al asignar un punto negro si el elemento de la matriz es un 1 y un punto blanco si el elemento de la matriz es un 0. Por lo tanto, los puntos negros corresponden a estados similares y los puntos blancos a estados diferentes.

Al calcular la gráfica de recurrencia en una serie temporal, en resumen, estamos comparando cada uno de los valores de una variable de estado en la solución numérica con los otros valores que dicha variable para distintos tiempos [11].

Las gráficas de recurrencia pueden ser utilizadas para entrenar algoritmos de aprendizaje supervisado. De esta manera, los algoritmos pueden aprender sobre la dinámica del acoplamiento de los procesos fisiológicos, y como resultado, realizar predicciones sobre la dinámica de sus variables de estado.

## 2.6. Aprendizaje automático (ML)

Aprendizaje automático (ML, por las siglas en inglés de *Machine Learning*) es una técnica que generalmente hace referencia a los cambios en los sistemas que realizan

tareas asociadas con la *inteligencia artificial* (IA), puede definirse como el uso de sistemas informáticos que no requieren programación explícita para aprender sobre la tarea que están completando [40].

El aprendizaje automático se divide en dos categorías principales: *aprendizaje no supervisado* y *aprendizaje supervisado* [40, 22, 28]. El aprendizaje no supervisado aprende las propiedades de los datos sin la necesidad de alguna guía humana, en otras palabras, el aprendizaje no supervisado es capaz de analizar datos sin necesidad de que los humanos los etiqueten explícitamente para identificar patrones específicos de información, ahorrando tiempo y recursos. Por otro lado, el aprendizaje supervisado requiere datos etiquetados para aprender la relación entre una entrada  $X$  y una salida  $Y$ , de esta manera, permite que el modelo se generalice y sea capaz de etiquetar de manera correspondiente los datos que se van a predecir, esto es, predecir las salidas  $Y$  para nuevos valores de  $X$ . Ejemplos de este tipo de aprendizaje son las regresiones tradicionales y el deep-learning. El aprendizaje supervisado requiere de la participación humana para etiquetar las entradas. Cada modelo de aprendizaje supervisado debe pasar por un proceso de entrenamiento y prueba para garantizar la generalización. Los modelos de aprendizaje supervisado se pueden dividir en dos grandes grupos: modelos de clasificación y modelos de regresión; en el primero (como en este estudio), la variable debe predecirse como variable cualitativa o categórica, cada variable es clasificada en una categoría denominada *valor de clase*. En los modelos de regresión, la variable a predecir es una variable cuantitativa, ya sea continua o discreta [28].

Las entradas  $X$  son comúnmente matrices con  $n \times n$  entradas, donde  $n$  es cada una de las características a estudiar de cada especie estudiada. Los valores de los componentes pueden ser de tres tipos principales: números con valores reales, números con valores discretos o valores categóricos [22]. Un ejemplo para ilustrar una entrada  $X$  con valores categóricos en una matriz columna de cuatro entradas, es el siguiente: la información sobre un estudiante puede estar representada por los valores de año que está cursando, materia, sexo, consejero. Entonces, un estudiante en particular estaría representado por un vector de la siguiente forma:

$$X = \begin{bmatrix} \text{primer año} \\ \text{física} \\ \text{hombre} \\ \text{Sánchez} \end{bmatrix} \quad (2.34)$$

En cambio, las salidas  $Y$  pueden ser de tres tipos: un número real, (en cuyo caso la función  $h$  que relaciona la salida con su correspondiente entrada, se denomina *función de estimación*, y a la salida se le denomina *estimación*) un valor categórico (en este caso, la función  $h$  es denominada *clasificador*, y a la salida se le llama *etiqueta o valor de clase, categoría o decisión*) o un vector con componentes que son números reales o valores categóricos.

Se le denomina *conjunto de entrenamiento* a un conjunto de entradas  $X$  con sus correspondientes salidas  $Y$  que el modelo de ML utilizará para aprender sobre el proceso que se desea realizar, y de esta manera, pueda obtener la función de relación  $h$  [40, 22, 28].

En este estudio de investigación, los valores de entrada son imágenes de las gráficas de recurrencia, las salidas son etiquetas de clase y buscamos una función  $h$  que clasifique estas imágenes.

Es importante resaltar que en un algoritmo ML, también existe ruido, y en este caso el ruido corrompe los vectores del conjunto de entrenamiento. Existen dos tipos diferentes de ruido: el ruido denominado de clase, que altera aleatoriamente el valor de la función, y el ruido denominado de atributo, el cual, altera aleatoriamente los valores de los componentes del vector de entrada [22].

Las aplicaciones de ML en el desarrollo tecnológico actual son bastantes, se extienden desde las áreas críticas y económicamente masivas como negocios de alta tecnología, manufactura, defensa nacional y cuidado de la salud, hasta el desarrollo de investigación para circunstancias más específicas como la ciencia de datos y la ciencia de materiales [29].

A continuación, se describen algunos de los modelos de aprendizaje supervisado más utilizados.

### 2.6.1. Modelos de aprendizaje supervisado

**Regresión logística:** El modelo de regresión logística consiste en calcular una probabilidad de pertenecer o no a una clase. Sea  $Y$  la variable dependiente que representa la pertenencia o no a una de las clases,  $Y$  toma valores de 1 si p

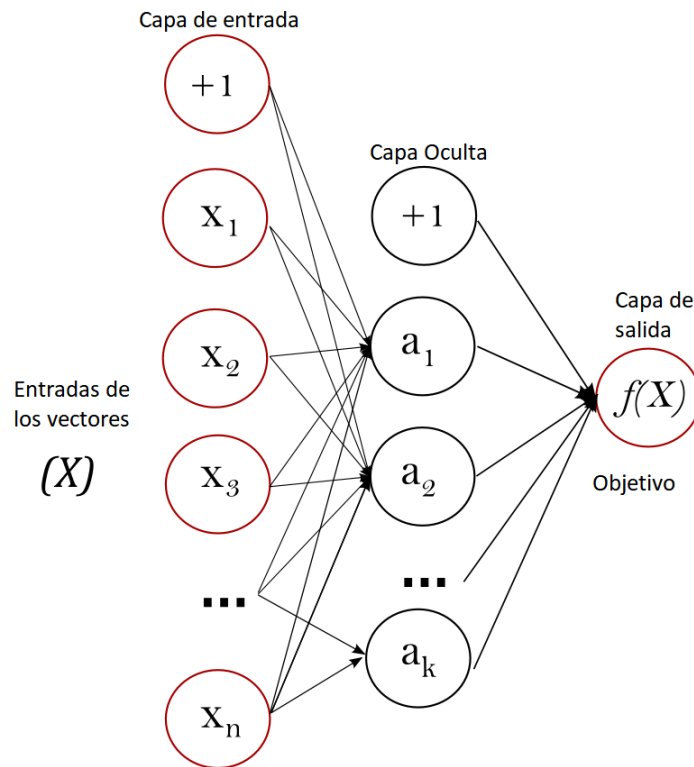
$$\Pr(Y = 1|x_1, x_2, \dots, x_p) = \frac{1}{1 + \exp(-\alpha - \beta_1 x_1 - \beta_2 x_2 - \dots - \beta_p x_p)} \quad (2.35)$$

en donde  $x_1, x_2, \dots, x_p$  son las entradas o descriptores de la variable a clasificar, y  $\alpha, \beta_1, \beta_2, \dots, \beta_p \in \mathbb{R}$  son parámetros del modelo. El procedimiento de estimación de estos parámetros se basa en el método de máxima verosimilitud, el cual consiste en la obtención del máximo del logaritmo de la función de verosimilitud:

$$L(y, \beta) = \sum_{i=1}^n y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i) \quad (2.36)$$

donde  $n$  es el número de observaciones y  $p_i = \Pr(Y = y_i|x_i)$  [38].

**Red neuronal Perceptrón Multicapa (MLP):** Un modelo de red neuronal utiliza el algoritmo de *perceptrón multicapa (MLP)*. Un perceptrón multicapa consiste en una red neuronal con una capa de entrada, una o más capas ocultas (generalmente solo una) y una capa de salida. Esta red es capaz de aprender una función  $f(\cdot) : \mathbb{R}^m \rightarrow \mathbb{R}^o$  donde  $m$  y  $n$  son las dimensiones del conjunto de datos de entrada y de salida correspondientemente, dado un conjunto de características  $X = x_1, x_2, \dots, x_m$  y un objetivo  $y = f(X)$  [33].



**Figura 2.1:** Perceptrón de una capa oculta. La capa más a la izquierda, conocida como capa de entrada, consta de un conjunto de neuronas  $\{x_i | x_1, x_2, \dots, x_m\}$  que representa a las entradas o descriptores de la variable a predecir. Cada neurona de la capa oculta transforma los valores de la capa anterior con una suma lineal ponderada  $w_1x_1 + w_2x_2 + \dots + w_mx_m$ , seguido de una función de activación no lineal  $g(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ . La capa de salida recibe los valores de la última capa oculta y los transforma en valores de salida.

El perceptrón puede aprender modelos no lineales para clasificación o regresión, mediante el proceso de retropropagación, el cual, consiste en el entrenamiento bajo aprendizaje supervisado, por lo tanto, el modelo necesita un conjunto de entrenamiento que le describa cada entrada y su valor de clase. Sea:  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ ; donde  $x_i \in \mathbb{R}^n$  corresponden a cada una de las gráficas de recurrencia y  $y_i \in \{0, 1\}$ .

Una neurona de la capa oculta del MLP aprende la función:

$$f(x) = W_2 g(W_1^T x + b_1) + b_2 \quad (2.37)$$

donde  $W_1 \in \mathbb{R}^m$  y  $W_2, b_1, b_2 \in \mathbb{R}$  son parámetros del modelo.  $W_1$  y  $W_2$  representan los pesos de la capa de entrada y la capa oculta respectivamente;  $b_1, b_2$  representan el sesgo agregado a la capa oculta y la capa de salida respectivamente.  $g(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$  es la función de activación.

A partir de pesos aleatorios iniciales, el perceptrón multicapa (MLP) minimiza la función de pérdida actualizando repetidamente estos pesos. Después de calcular la pérdida, una pasada hacia atrás la propaga desde la capa de salida a las capas anteriores, proporcionando a cada parámetro de peso un valor de actualización destinado a disminuir la pérdida.

Una función de pérdida comúnmente utilizada es la entropía cruzada:

$$Loss(\hat{y}, y, W) = -y \ln(\hat{y}) - (1 - y) \ln(1 - \hat{y}) + \alpha \|W\|_2^2 \quad (2.38)$$

donde  $\alpha \|W\|_2^2$  es un término de regularización L2 que penaliza modelos complejos;  $\alpha$  es un hiperparámetro no negativo que controla la magnitud de la penalización.

El algoritmo se detiene cuando alcanza un número máximo preestablecido de iteraciones; o cuando la mejora en la pérdida está por debajo de cierto número pequeño [34].

**kNN:** El algoritmo kNN (k-Nearest Neighbors) consiste en buscar un número  $k$  de ejemplos de entrenamiento más cercanos en el espacio de características de los descriptores de imagen y usa su promedio como predicción.

Los ejemplos de entrenamiento son vectores en el espacio característico multidimensional, cada vector cuenta con un número de entradas, aquí denominados atributos [25].

El espacio es particionado en regiones por localizaciones y etiquetas de clase según los datos de entrenamiento. Un punto en el espacio es asignado a la clase  $C$  si esta es la clase más frecuente entre los  $k$  datos de entrenamiento más cercano. La distancia que este algoritmo utiliza es la distancia euclidiana:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^p (x_{ri} - x_{rj})^2} \quad (2.39)$$

**SVM:** La máquina de vectores de soporte (SVM) es una técnica de aprendizaje

automático, la cual, separa el espacio de atributos con un hiperplano, maximizando así el margen entre los diferentes valores de clase.

Dado un conjunto de entrenamiento de pares vector-etiqueta  $(x_i, y_i)$ ,  $i = 1, \dots, l$  donde  $x_i \in \mathbb{R}^n$  y  $y \in 1, -1^l$ , las máquinas de vectores de soporte (SVM) requieren la solución del siguiente problema de optimización:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i \quad (2.40)$$

sujeto a  $y_i(w^T \varphi(x_i) + b) \geq 1 - \xi_i$ , y  $\xi_i \geq 0$ . Aquí los vectores de entrenamiento  $x_i$  se mapean en un espacio dimensional superior (tal vez infinito) por la función  $\varphi$ . SVM encuentra un hiperplano de separación lineal con el margen máximo en este espacio dimensional superior.  $C > 0$  es el parámetro de penalización del término de error. Además,  $K(x_i, x_j) \equiv \varphi(x_i)^T \varphi(x_j)$  se denomina la función del núcleo. La función del núcleo utilizada se conoce como radial basis function (RBF) y se define de la siguiente manera:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0 \quad (2.41)$$

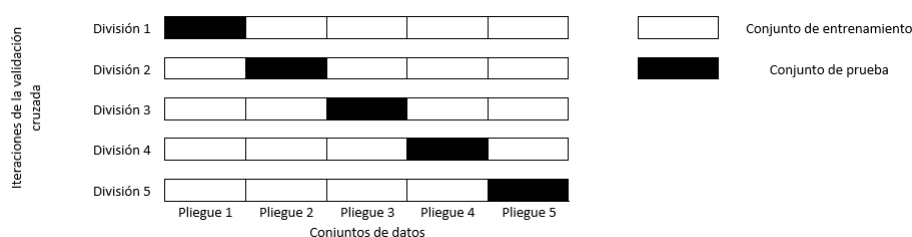
donde  $\gamma$  es un parámetro del núcleo [18].

## 2.6.2. Técnicas de entrenamiento de aprendizaje supervisado

Para realizar el entrenamiento de los modelos de ML supervisado, el conjunto de datos se divide en un conjunto de entrenamiento y uno de prueba el cual será clasificado en cada etiqueta de clase por cada modelo predictivo y se calculará la fracción de muestras clasificadas correctamente. Es importante recalcar que la razón por la que dividimos nuestros datos es que estamos interesados en medir qué tan bien se generaliza nuestro modelo a datos nuevos, no qué tan bien nuestro modelo se ajusta al conjunto de entrenamiento, es decir, conocer qué tan correctas son las predicciones del modelo para datos que no se observaron durante el entrenamiento. Ejemplos de técnicas de entrenamiento son la *validación cruzada*, la *validación cruzada estratificada* y la *técnica leave one out* [30].

La validación cruzada (simple) es un método estadístico para evaluar el rendimiento de la generalización de un método predictivo. El proceso consiste en dividir los datos repetidamente y así entrenar los modelos predictivos. Primero, se especifica el número  $k$  de veces en la que la validación cruzada será realizada. Cuando se

ejecuta una validación cruzada de  $k$  veces, los datos se dividen primero en  $k$  partes de (aproximadamente) igual tamaño, denominados pliegues. Los modelos son entrenados utilizando el primer pliegue como conjunto de prueba y los pliegues restantes (del segundo hasta el último,  $k - 1$  pliegues) se utilizan como conjunto de entrenamiento. Luego se evalúa la exactitud resultante en el primer pliegue, la *exactitud* consiste en la fracción de muestras clasificadas correctamente entre el total de muestras. Después los modelos nuevamente son entrenados pero esta vez usando el segundo pliegue como conjunto de prueba y los datos en los pliegues 1, 3, 4, ...,  $k$  como conjunto de entrenamiento y se vuelve a evaluar la exactitud. Este proceso se repite utilizando los pliegues siguientes hasta llegar a  $k$  como conjunto de prueba y los restantes como conjunto de entrenamiento. Para cada una de estas divisiones de los datos en conjuntos de entrenamiento y prueba, se calcula la exactitud. Al final, hemos recopilado  $k$  valores de exactitud para cada modelo predictivo. Por último, se calcula el promedio de estos valores y el promedio es la exactitud del modelo predictivo.



**Figura 2.2:** División de datos en cinco validaciones cruzadas (validación cruzada de 5 pliegues).

Ahora dado que las gráficas de recurrencia están organizados por etiqueta de clase con aproximadamente el mismo número de diagramas, los resultados no serán los óptimos al realizar una validación cruzada común, imaginando por ejemplo una validación cruzada de 3 pliegues, la exactitud de cualquier modelo predictivo será muy cercana a cero; esto debido a que al dividir los datos en 3 pliegues, cada pliegue corresponderá a casi el total de cada una etiqueta de clase y unos cuantos diagramas de otra, por lo que al dividir los datos en los conjuntos de entrenamiento y prueba, corresponderán casi a una etiqueta de clase como conjunto de prueba y las dos restantes al conjunto de entrenamiento, así entrenaremos los modelos con las gráficas de recurrencia correspondientes a los acoplamientos de la clase restante con unos pocos de la clase de prueba, por lo que nuestros modelos no llevarán a cabo un entrenamiento adecuado.

Por esta razón, en lugar de realizar una validación cruzada, el proceso óptimo es una *validación cruzada estratificada*. En la validación cruzada estratificada, dividimos los datos de manera que las proporciones entre clases, sean las mismas en cada pliegue que en todo el conjunto de datos, es decir, cada clase representa aproximadamente el 33.3% de muestras del total de datos, la validación cruzada estratificada asegura

que cada pliegue cuente con el 33% de datos de cada una de las clases, por lo tanto cada pliegue contará con aproximadamente el mismo número de gráficas de recurrencia de cada tipo de acoplamiento logrando que los modelos predictivos realicen un entrenamiento adecuado generalizando el proceso.

Otra forma de evaluar los resultados, es a través del método “*leave one out*”, el cual consiste en realizar una validación cruzada con el número de pliegues igual al número de muestras, de esta manera, se presenta un dato a la vez, y el modelo es entrenado con todos los demás datos restantes; después se procede a predecir la clase del segundo dato, entrenando al algoritmo con todos los datos restantes, así sucesivamente hasta terminar con todo el conjunto.

### 2.6.3. Métricas de evaluación

Cuando un modelo predictivo toma una decisión en la clasificación de uno de los datos etiquetados, desde el punto de vista de cada etiqueta de clase, pueden ocurrir 4 diferentes escenarios [30]. Si suponemos que hacemos la clasificación sobre la clase  $A$  y existen además otras dos etiquetas  $B$  y  $C$ , se presentan los siguientes escenarios:

1. Que uno de los datos correspondiente a la clase  $A$  sea correctamente clasificado en la clase  $A$ , esta decisión se denomina un *verdadero positivo* ( $TP$ , por sus siglas en inglés).
2. Que uno de los datos no correspondiente a la clase  $A$  (cualquier dato de las clases  $B$  o  $C$ ) sea incorrectamente clasificado en la clase  $A$ , esta decisión se denomina un *falso positivo* ( $FP$ ).
3. Que un dato no correspondiente a la clase  $A$  (nuevamente, cualquiera de  $B$  o  $C$ ) sea clasificado en una clase que no sea  $A$  (en las clases  $B$  o  $C$ ), esta decisión se denomina un *verdadero negativo* ( $TN$ ).
4. Que un dato correspondiente a la clase  $A$  sea clasificado en una clase que no sea  $A$  (puede ser  $B$  o  $C$ ) esta decisión es un *falso negativo* ( $FN$ ).

Es de notar que la clasificación de muestras como verdaderos negativos no implica que sean verdaderos positivos desde el punto de vista de las demás clases. Por ejemplo, un dato correspondiente a la clase  $B$ , desde la perspectiva de la clase  $A$ , si se clasifica en cualquier otra clase que no sea  $A$  misma, es un verdadero negativo; pero si no es clasificado en la clase  $B$  no será un verdadero positivo para la clase  $B$ .

Estos resultados son resumidos a través de la *matriz de confusión*. La matriz de confusión es una matriz con las proporciones entre la clase prevista por cada modelo y la real. Las filas representan a la clase real y las columnas la clase predicha por los modelos probabilísticos. La diagonal superior de esta matriz corresponde a los datos correctamente clasificados; un puntaje de 1.0 en todas las métricas corresponde a una matriz de confusión diagonal [30].

		Predicha			
		A	B	C	$\Sigma$
Real	A	44	9	0	53
	B	9	40	6	55
	C	2	3	46	51
$\Sigma$		59	48	52	159

**Cuadro 2.2:** Ejemplo de matriz de confusión para tres clases A, B y C; cada clase cuenta con 53, 55 y 51 datos correspondientemente. Las filas representan la clase real de cada dato a predecir, mostrando el número de datos reales por clase en la última entrada de la fila; mientras que las columnas representan las decisiones tomadas por los modelos de aprendizaje automático, en este caso, la última representa el número de datos por clase según los modelos predictivos.

Bajo las cuatro posibilidades en la clasificación, podemos construir métricas relativas a cada variable de clase que nos permiten conocer más acerca de qué tan adecuado es un modelo predictivo, dado que la exactitud no siempre es una medida determinante.

Considerando los cuatro escenarios descritos arriba, la *exactitud* (CA, classification accuracy) queda determinada de la siguiente forma:

$$CA = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.42)$$

La exactitud es una medida de qué tan correctas son las predicciones realizadas.

La *precisión* mide cuántas de las muestras predichas como positivas son realmente positivas y es utilizada como métrica de rendimiento cuando el objetivo es limitar el número de falsos positivos.

$$\text{Precisión} = \frac{TP}{TP + FP} \quad (2.43)$$

La tasa de verdaderos positivos denominada como *recall*, por otro lado, mide cuántas de las muestras positivas son capturadas por las predicciones positivas. El recall se utiliza como métrica de rendimiento cuando necesitamos identificar todas las muestras positivas; es decir, cuando es importante evitar falsos negativos.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2.44)$$

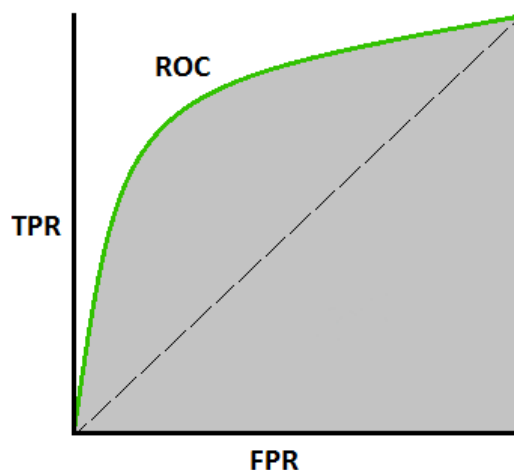
Si bien la precisión y el recall son medidas muy importantes, mirar solo una de ellas no proporciona una imagen completa del panorama completo. Una forma de resumirlos es la puntuación F1:

$$F1 = \frac{\text{precisión} * \text{recall}}{\text{precisión} + \text{recall}} \quad (2.45)$$

La última métrica que usaremos es el *área de bajo de la curva ROC (AUC)*. Una curva ROC (Curva Característica Operativa del Receptor) es un gráfico que muestra el rendimiento de un modelo de clasificación en todos los umbrales de clasificación posibles, un umbral de clasificación, se define como el valor al cual un modelo predictivo toma la decisión de clasificar una muestra en una categoría; un valor por encima del umbral indica que pertenece a la clase, mientras que un valor por debajo del umbral indica que no. Una curva ROC representa dos parámetros:

- La tasa de verdaderos positivos:  $TPR = \text{Recall} = \frac{TP}{TP+FN}$
- La tasa de falsos positivos:  $FPR = \frac{FP}{FP+TN}$

La curva ROC, grafica TPR contra FPR, donde TPR está en el eje  $y$  y FPR está en el eje  $x$ , en diferentes umbrales de clasificación. Reducir el umbral de clasificación, en general, clasifica más elementos como positivos, por lo que aumentarán tanto los falsos positivos como los verdaderos positivos. La curva ideal está cerca de la parte superior izquierda, se desea un clasificador que produzca un recall alto y mantenga una tasa baja de falsos positivos.



**Figura 2.3:** Ejemplo gráfico de una curva ROC característica. El AUC se representa con la parte sombreada.

De esta manera, el AUC mide el área bidimensional por debajo de la curva ROC. La curva AUC es una medida de rendimiento para los problemas de clasificación en

varios valores del umbral representando el grado o medida de separabilidad. Indica cuánto es capaz el modelo de distinguir entre clases. Cuanto mayor sea el AUC, mejor será el modelo para predecir una clase de las demás [10].

Por último, es posible emplear una matriz para comparar los modelos predictivos entre sí, dependiendo el puntaje obtenido en cada métrica. La matriz muestra las probabilidades de que la puntuación del modelo de la fila sea mayor que la del modelo de la columna. A esta matriz se le conoce como *Matriz Bayesiana*.

Para la construcción de esta matriz, se compara estadísticamente alguna métrica de dos modelos predictivos que se han evaluado mediante  $m$  corridas de validación cruzada en  $k$  pliegues. Proporcionamos a ambos clasificadores los mismos conjuntos de entrenamiento y prueba y calculamos la diferencia en los resultados de la métrica. Esto produce las diferencias de resultados  $\vec{x} = (x_1, x_2, \dots, x_n)$ , donde  $n = mk$ . Denotamos la media muestral y la varianza muestral de las diferencias como  $\bar{x}$  y  $\hat{\sigma}^2$ . Una prueba estadística que compare a los modelos tiene que establecer si la diferencia media entre los dos clasificadores es significativamente diferente de cero, analizando el vector de resultados  $\vec{x}$ , así se define el estadístico que correlaciona a los modelos predictivos  $t$  de la siguiente manera:

$$t = \frac{\bar{x}}{\sqrt{\hat{\sigma}^2 \left( \frac{1}{n} + \frac{n_{te}}{n_{tr}} \right)}} \quad (2.46)$$

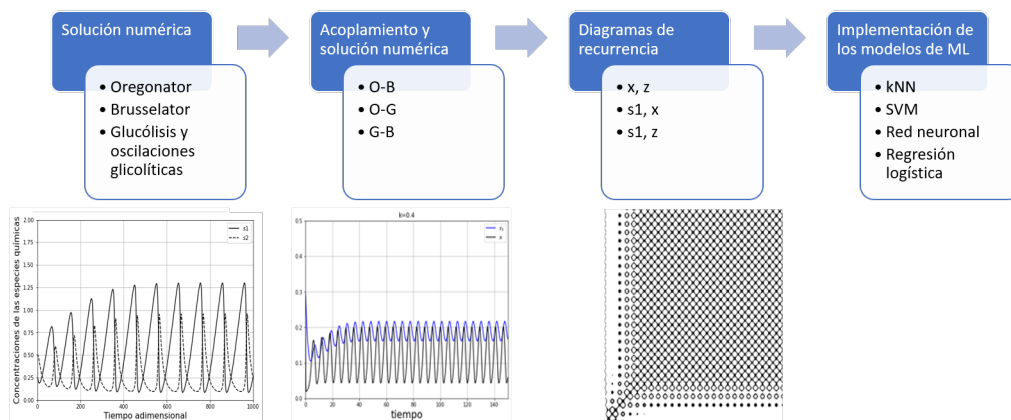
donde  $n_{te}$ ,  $n_{tr}$  y  $n_{tot}$  denotan el tamaño del conjunto de entrenamiento, del conjunto de prueba y de todo el conjunto de datos disponibles [5].

Esta matriz solo puede construirse cuando el modelo es entrenado y evaluado por el método de validación cruzada estratificada y no por leave one out.



### 3 Metodología

La metodología se dividió en cuatro partes, la primera consiste en la solución numérica de cada modelo por separado, la segunda consistió en la solución numérica de los modelos acoplados, la tercera en la obtención de las gráficas de recurrencia correspondientes a cada acoplamiento y la última la implementación de los modelos de aprendizaje automático para la clasificación de cada diagrama.



**Figura 3.1:** Diagrama de flujo que muestra la metodología implementada. Primero se calculan las soluciones numéricas para cada uno de los tres modelos, después, son acoplados entre sí (oregonator con brusselator O-B, oregonator con glucólisis O-G y glucólisis con brusselator G-B) y nuevamente se obtiene la solución numérica del sistema. El resultado de las soluciones numéricas, son en efecto, series temporales para cada una de las variables acopladas ( $x$  y  $z$  en O-B,  $\sigma_1$  y  $x$  en O-G y por último  $\sigma_1$  y  $z$  en G-B), por lo que, es posible calcular la gráfica de recurrencia para cada una de estas series y por consiguiente, entrenar modelos de aprendizaje automático.

## 3.1. Soluciones numéricas

Los modelos, se resolvieron numéricamente empleando el método de Runge-Kutta de cuarto orden con  $h = 1$  (ecuación 2.9) en el lenguaje de programación Python V3.8, para un intervalo de tiempo de 0.01 s.

### 3.1.0.1. Oregonator

La dinámica del primer modelo está determinada por el siguiente par de ecuaciones diferenciales:

$$\begin{aligned} \varepsilon \frac{dx}{dt} &= x(1-x) + f \frac{q-x}{q+x} z \\ \frac{dz}{dt} &= x - z \end{aligned} \quad (3.1)$$

Los parámetros empleados para la solución numérica del modelo del oregonator se muestran en el cuadro 3.1.

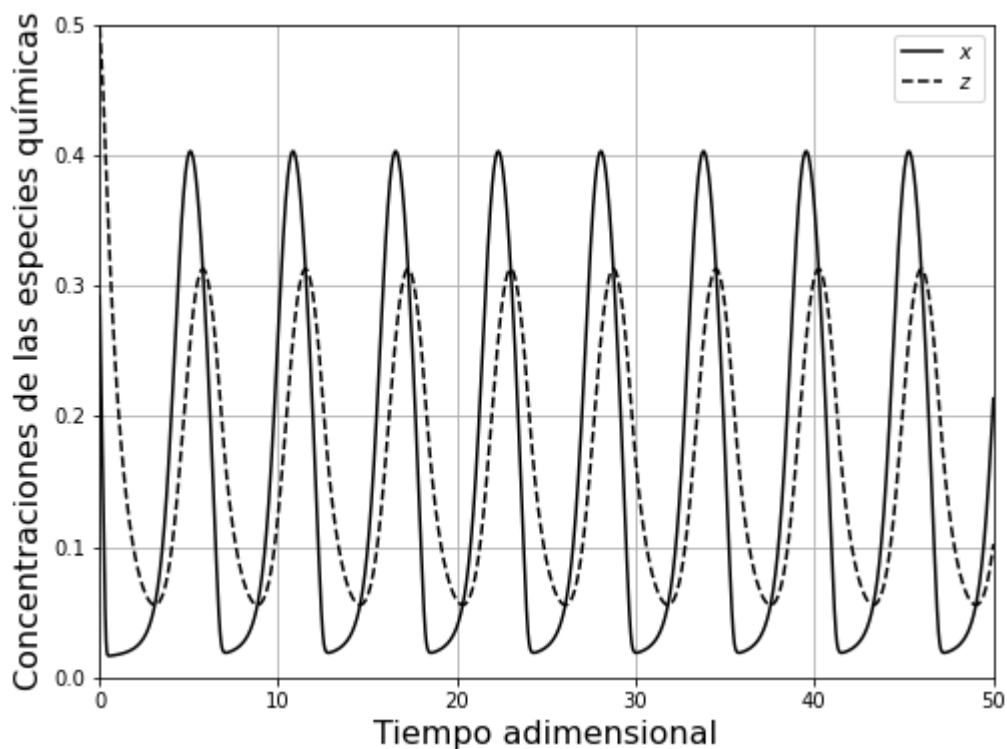
**Cuadro 3.1:** Tabla de valores de los parámetros adimensionales en el modelo oregonator.

Parámetro	Valor
$\varepsilon$	0.3
$q$	0.015
$f$	1

De esta manera, la solución numérica, con valores iniciales  $x_0 = 0.3$  y  $z_0 = 0.5$ , se muestra en la figura 3.2.

Notar que el papel de activador representa a una especie autocatalítica, es decir, sus productos desempeñan el papel de catalizadores para la misma especie  $x$ , para que después, la especie  $x$  sea regulada por  $z$ , disminuyendo la actividad catalítica.

En la figura 3.2 podemos notar que a medida que  $x$  crece, la variable  $z$  también crece y comienza a frenar la producción de  $x$  hasta que su concentración llega a un punto máximo aproximadamente en  $x_{m\acute{a}x} \simeq 0.402$ , para después comenzar a disminuir su concentración debido a que  $z$  sigue creciendo hasta el punto en que ambas concentraciones son iguales  $z_{m\acute{a}x} \simeq 0.314$ , de esta manera  $z$  comienza a disminuir también. A medida que  $x$  decrece,  $z$  también decrece hasta la concentración mínima  $x_{m\acute{i}n} \simeq 0.025$ , ahora  $z$  ha disminuido tanto que permite que la autocatálisis



**Figura 3.2:** Solución numérica para el modelo oregonator podemos notar el comportamiento oscilatorio para las dos variables donde la especie  $x$  desempeña el papel de activador y la especie  $z$  el papel de inhibidor.

de  $x$  aumente su concentración hasta que  $z$  llega a su punto mínimo en  $z_{min} \simeq 0.068$  y el ciclo vuelve a comenzar.

### 3.1.0.2. Brusselator

La dinámica del segundo modelo está determinada por el siguiente par de ecuaciones diferenciales:

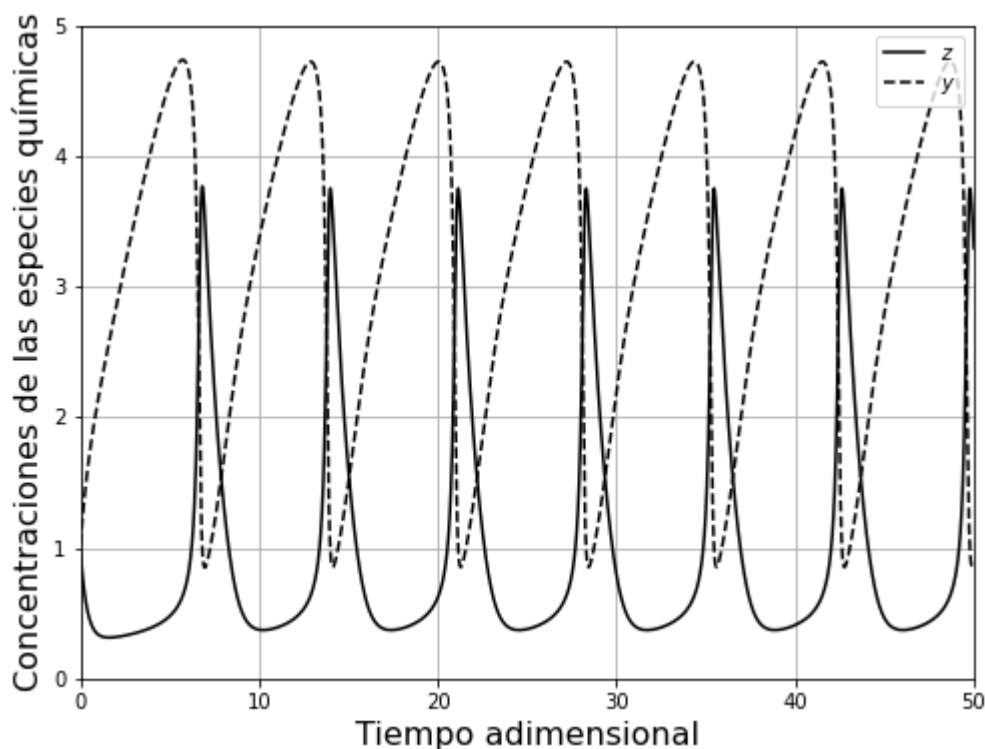
$$\begin{aligned} \frac{dz}{dt} &= 1 - (b + 1)z + ayz^2 \\ \frac{dy}{dt} &= bz - ayz^2 \end{aligned} \quad (3.2)$$

los valores de los parámetros para este modelo se muestran en el cuadro 3.2.

**Cuadro 3.2:** Tabla de valores de los parámetros en el modelo brusselator.

Parámetro	Valor
$a$	1.0
$b$	3.0

La solución numérica con valores iniciales  $z_0 = 1.0$  y  $y_0 = 1.0$  se muestra en la figura 3.3.



**Figura 3.3:** Solución numérica para el modelo brusselator. Nuevamente se presenta el comportamiento oscilatorio típico de una reacción autocatalítica. En este caso, la especie  $z$  desempeña el papel de activador y la especie  $y$  el de inhibidor.

La figura 3.3 nos permite notar que a medida que  $z$  crece, la concentración de  $y$  decrece y comienza a frenar la producción de  $z$  (de manera contraria a la dinámica del oregonator). La variable  $z$  crece hasta que su concentración llega a un punto máximo aproximadamente en  $x_{m\acute{a}x} \simeq 3.708$ , para después comenzar a disminuir su concentración. Por otro lado,  $y$  disminuye hasta llegar  $y_{m\acute{i}n} \simeq 0.894$ , y en ese tiempo,  $z$  comienza a disminuir. A medida que  $z$  decrece,  $y$  aumenta, hasta la concentración

máxima de  $z_{\min} \simeq .375$ , y  $y_{\max} \simeq 4.771$  y el ciclo vuelve a comenzar.

### 3.1.0.3. Glucólisis y oscilaciones glicolíticas

La dinámica del último modelo está determinada por el siguiente par de ecuaciones diferenciales:

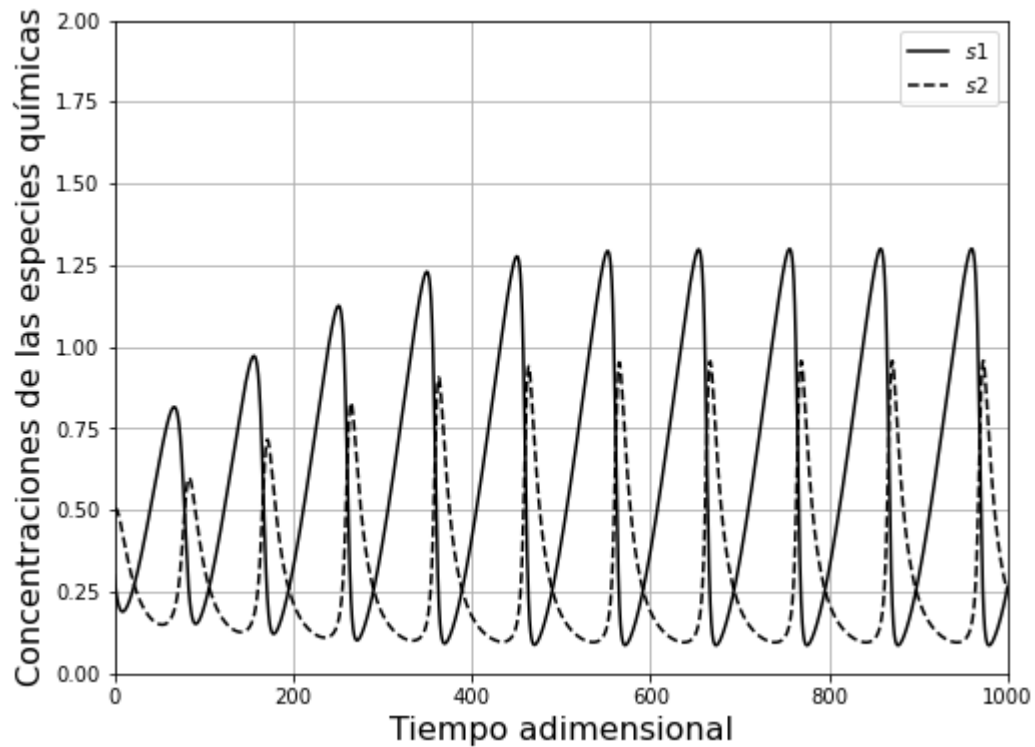
$$\begin{aligned} \frac{d\sigma_1}{d\tau} &= v - \frac{\sigma_1\sigma_2^\gamma}{\sigma_2^\gamma\sigma_1 + \sigma_2^\gamma + 1} \\ \frac{d\sigma_2}{d\tau} &= \alpha \frac{\sigma_1\sigma_2^\gamma}{\sigma_2^\gamma\sigma_1 + \sigma_2^\gamma + 1} - \eta\sigma_2 \end{aligned} \tag{3.3}$$

Como vimos en el apartado 2.4.3,  $\sigma_1$  y  $\sigma_2$  son parámetros adimensionales relacionados directamente con la concentración de ATP y ADP respectivamente. El valor de los parámetros de dicho modelo se aprecia en el cuadro 3.3.

**Cuadro 3.3:** Tabla de valores de los parámetros adimensionales en el modelo de glucólisis.

Parámetro	Valor
$v$	0.0285
$\eta$	0.1
$\alpha$	1
$\gamma$	2

Con los valores iniciales  $\sigma_{1,0} = 0.3$  y  $\sigma_{2,0} = 0.5$ , la solución es:



**Figura 3.4:** Solución numérica para el modelo de glucólisis mostrando la evolución de  $\sigma_1$  y  $\sigma_2$  (en el gráfico aparecen como  $s_1$  y  $s_2$  respectivamente) para el modelo de glucólisis de Sel'kov hacia una solución periódica.

Podemos notar que para este modelo las oscilaciones son más lentas, denotando así, un ciclo de retroalimentación positiva justo como vimos en el apartado 2.4.3.

### 3.1.1. Soluciones numéricas de los modelos acoplados

En la sección 1.3 vimos que es posible construir una función de acoplamiento de la siguiente manera:  $c(\mathbf{x}, \mathbf{x}_s) = \mathbf{k}(\mathbf{x}) \cdot (\mathbf{x} - \mathbf{x}_s)$ , donde  $\mathbf{x}$  y  $\mathbf{x}_s$  son los vectores de coordenadas del sistema maestro y esclavo respectivamente.

Esta función puede simplificarse de dos maneras, la primera es sincronizar en una sola dirección, esto significa que los vectores  $\mathbf{x}$  y  $\mathbf{x}_s$  pasen a ser solo una de las coordenadas  $x_i$  y  $x_{s,j}$  de los sistemas; en otras palabras, se debe seleccionar una de las variables de estado del sistemas maestro y una de las variables de estado del sistema esclavo y sustituirlas en sus vectores correspondientes. La segunda forma es elegir  $k = \text{constante}$  (*acoplamiento lineal*). De esta manera:

$$c(x_i, x_{s,j}) = k(x_i - x_{s,j}) \quad (3.4)$$

Es importante recalcar que esos sistemas fueron elegidos, porque son tres sistemas típicos que presentan oscilaciones y están relacionados con sistemas fisiológicos.

Se realizó el acoplamiento bidireccional lineal de los siguientes sistemas: *Oregonator con brusselator (O-B)*, *oregonator con glucólisis y oscilaciones glicolíticas (O-G)* y *glucólisis y oscilaciones glicolíticas con brusselator (G-B)*; con valores del parámetro de sincronización  $k$ : 0.1 a 1.1 en intervalos de 0.1 dado que con estos valores, las nuevas soluciones de los sistemas acoplados presentan oscilaciones y .

#### 3.1.1.1. Oregonator con brusselator (O-B)

El modelo oregonator puede utilizarse para entender la dinámica del ciclo de Krebs, por ello con la simulación del primer acoplamiento O-B, se estudia de manera análoga la sincronización fisiológica entre el ciclo de Krebs. El ciclo de Krebs o también conocido como ciclo del ácido cítrico, es una ruta metabólica que forma parte de la respiración celular de las células aeróbicas; toma acetil (CoA) producida por la oxidación del piruvato ( $C_3H_4O_3$ ) como materia prima y en una serie de reacciones, recolecta gran parte de la energía de sus enlaces formando moléculas de NADH,  $FADH_2$  y ATP. Las primeras dos moléculas son conocidas como acarreadores de electrones reducidos, los cuales, tienen como objetivo ceder sus electrones a la cadena de transporte de electrones y mediante fosforilación oxidativa, genera la mayor parte del ATP producido en la respiración celular [46].

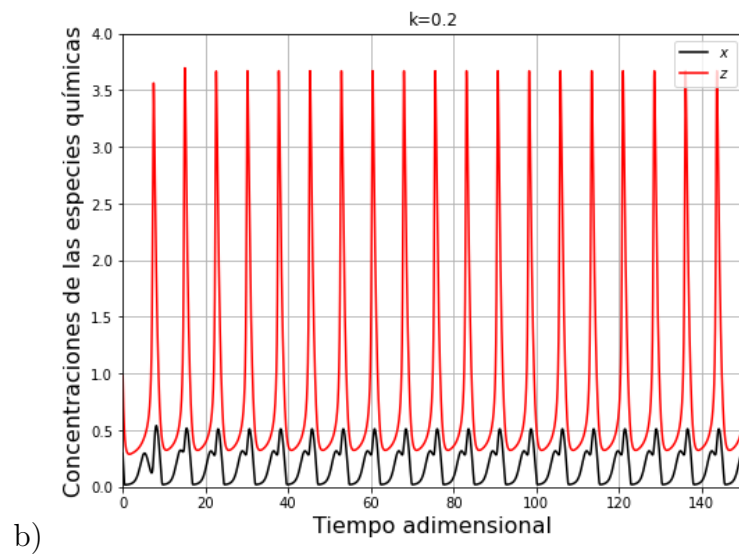
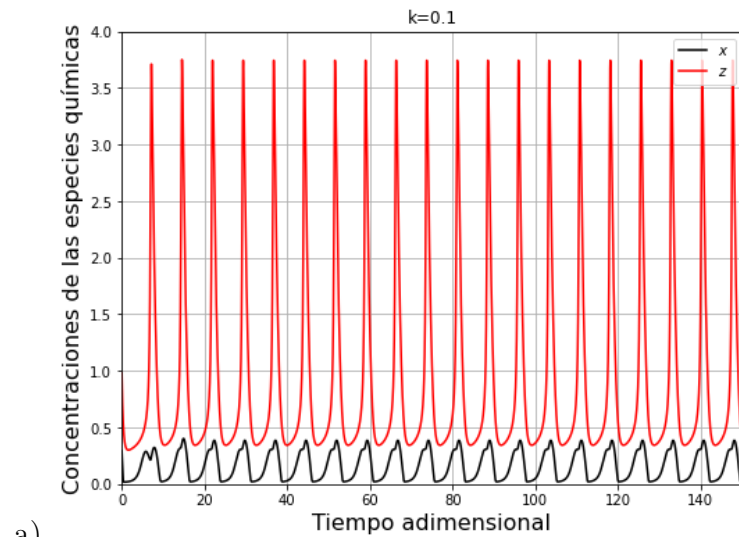
Por otro lado, el sistema brusselator representa un modelo útil para el estudio de la dinámica de procesos cooperativos químicos, en específico aquellos problemas que involucren lidiar con reacciones químicas con dos variables intermedias y una serie de productos iniciales y finales cuyas concentraciones están controladas durante todo el proceso de reacción. Un ejemplo de estos procesos, es la reacción trimolecular que surge en la formación de ozono por el oxígeno atómico a través de una colisión triple. Surge también en reacciones enzimáticas de carácter autocatalítico y en la física del plasma y del láser en múltiples acoplamientos entre ciertos modos [1]. Por lo que este acoplamiento puede ser la representación de la sincronización del ciclo de Krebs con alguna reacción trimolecular cuya dinámica sea regida por el modelo brusselator.

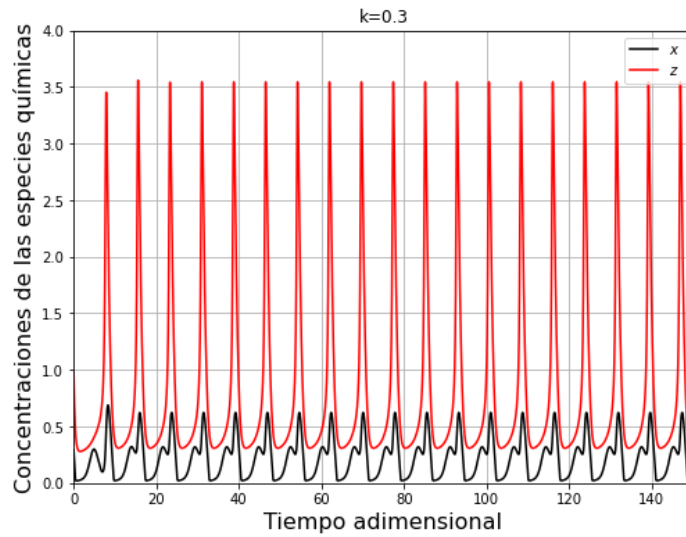
Algunas patologías asociadas a problemas con el ciclo de Krebs son el síndrome de Leigh, deficiencia del complejo II de la cadena respiratoria mitocondrial, susceptibilidad a los paragangliomas de cabeza y cuello, y susceptibilidad al carcinoma de células renales de aparición temprana [35, 16]. Por lo tanto, ambos ciclos están presentes y acoplados en la vida de las células aeróbicas y su desarrollo [39].

Las figuras 3.5 muestran las soluciones numéricas del acoplamiento O-B (la especie  $x$  del oregonator se muestra en negro mientras que la especie  $z$  perteneciente al brusselator en rojo). Lo primero a notar es que a medida de que  $k$  aumenta, la amplitud en la solución numérica de  $x$  aumenta también, mientras que la amplitud de  $z$  disminuye; esto debido a que el acoplamiento se vuelve más efectivo a medida que

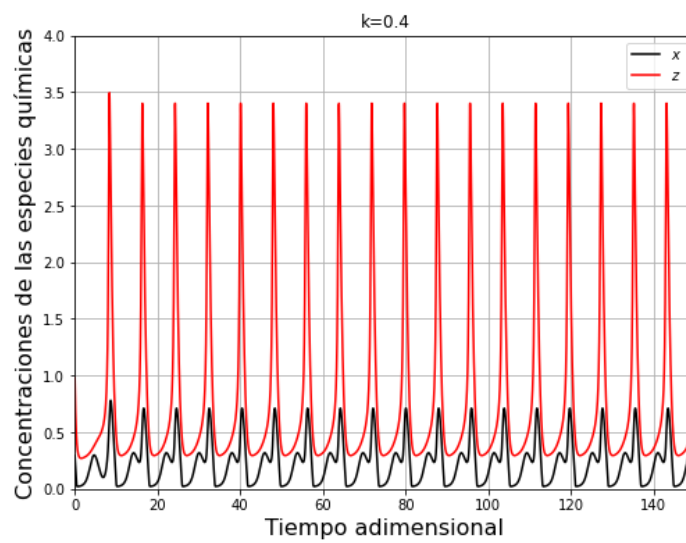
$k$  aumenta. También podemos notar un acoplamiento en la frecuencia de oscilación de  $x$  y  $z$  desde que el primer valor de  $k$ .

También es notoria una pequeña meseta (punto de inflexión) en la solución  $z$  que se vuelve más pronunciada a medida que  $k$  aumenta, esto debido a que para valores más grandes de  $k$ , fuera de nuestro rango de estudio,  $x$  y  $z$  presentarán oscilaciones más frecuentes y cada meseta formará un nuevo ciclo de oscilación.

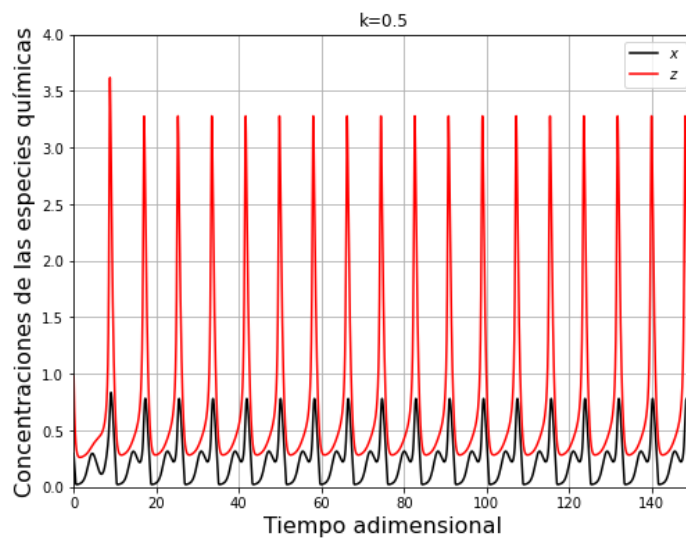




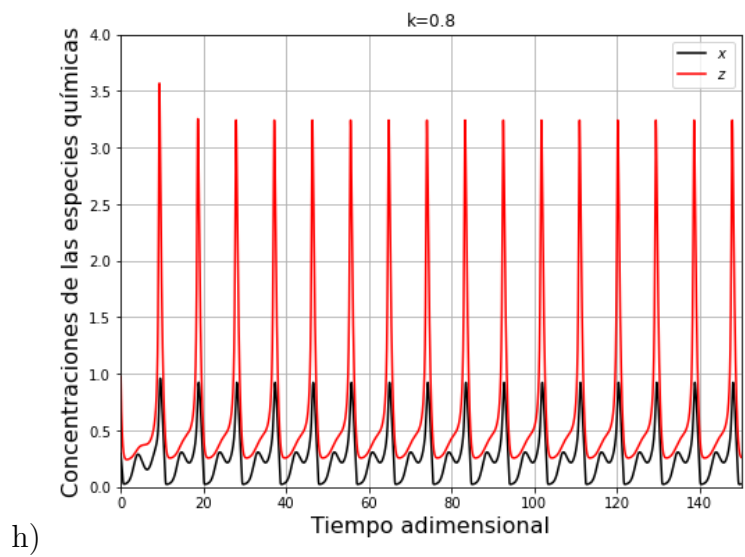
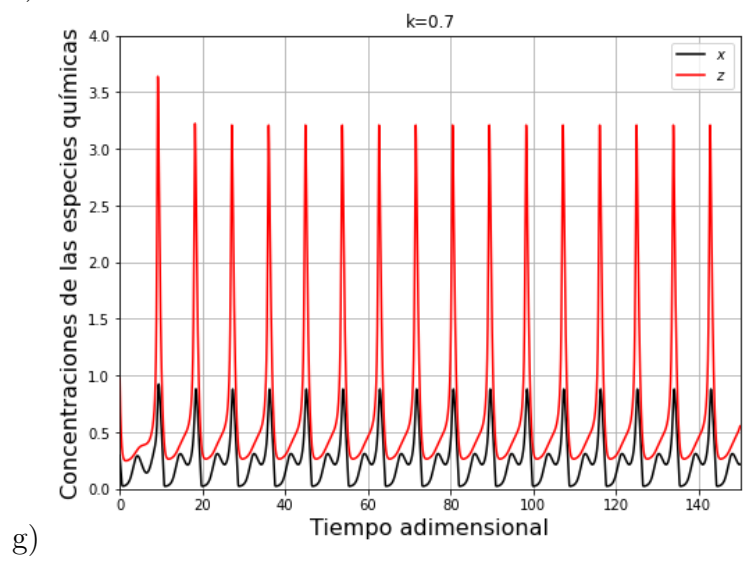
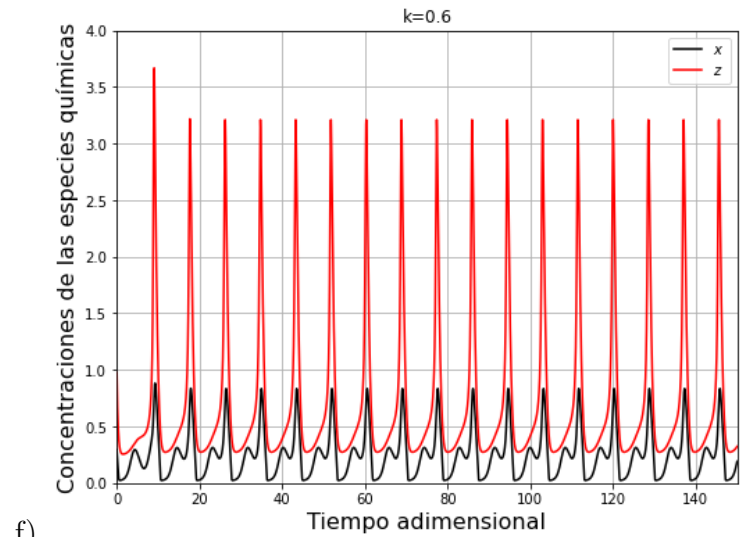
c)

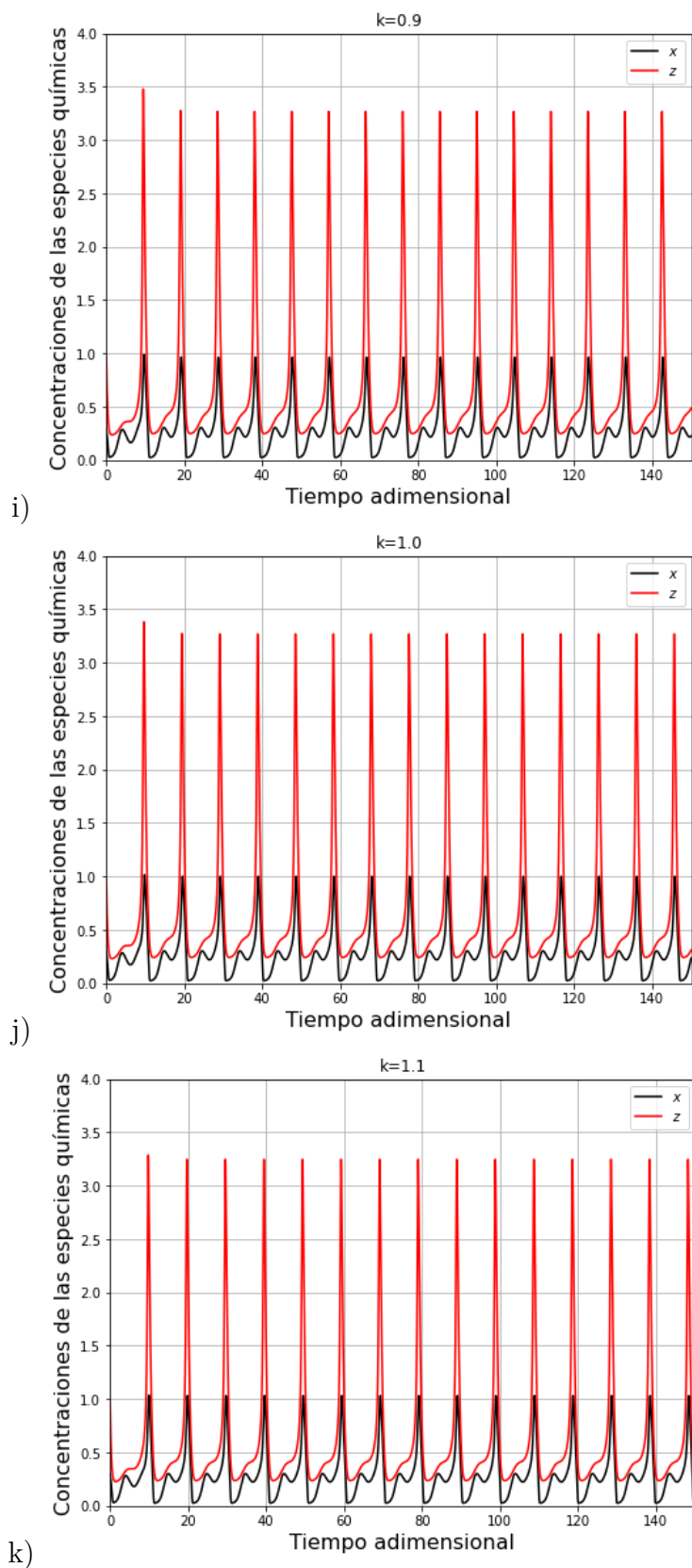


d)



e)





**Figura 3.5:** Solución numérica para el acoplamiento bidireccional lineal de los sistemas oregonator y brusselator para valores de  $k = 0.1, 0.2, \dots, 1.1$ .

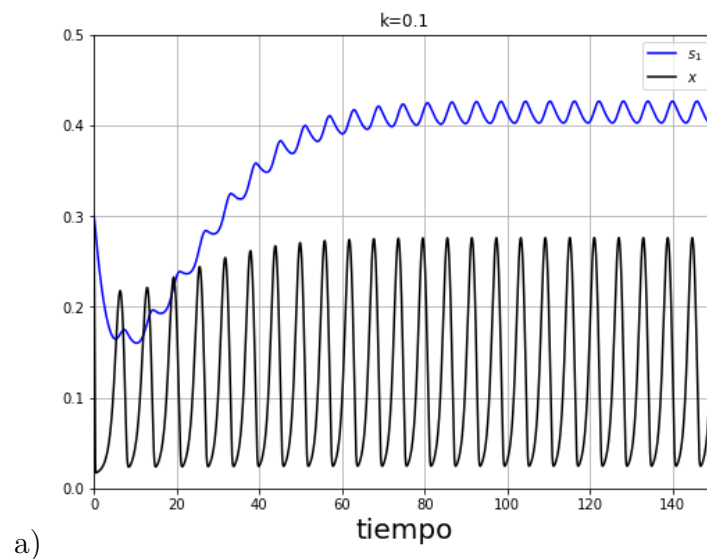
### 3.1.1.2. Oregonator con glucólisis y oscilaciones glicolíticas (O-G)

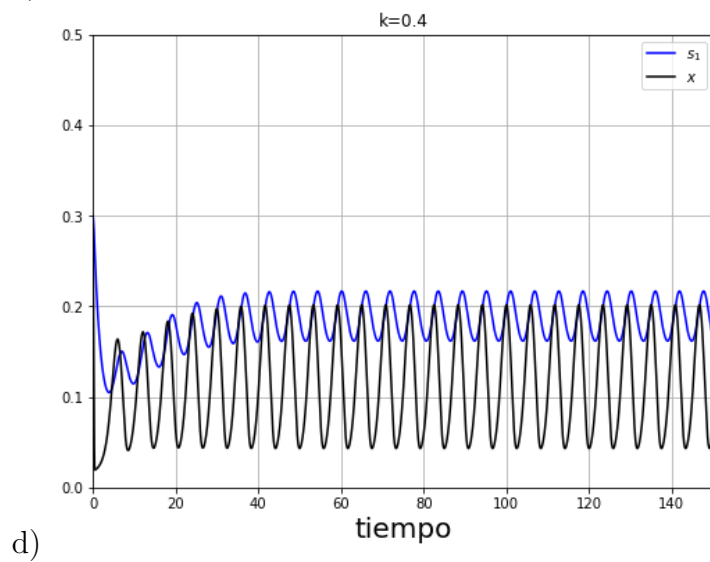
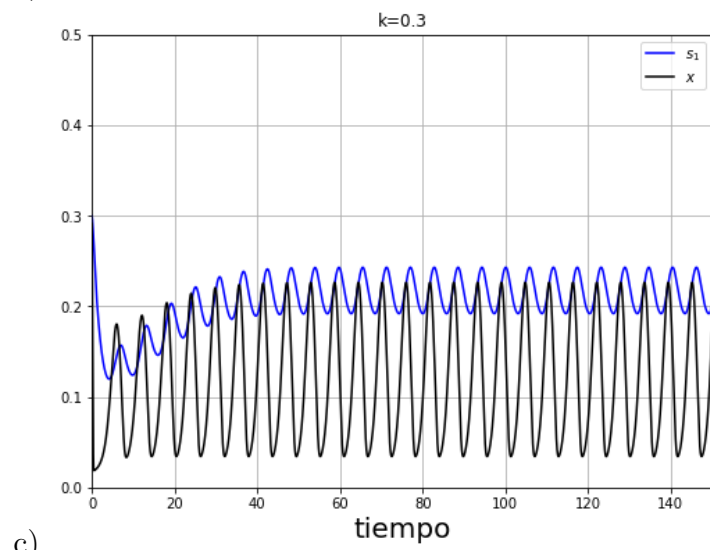
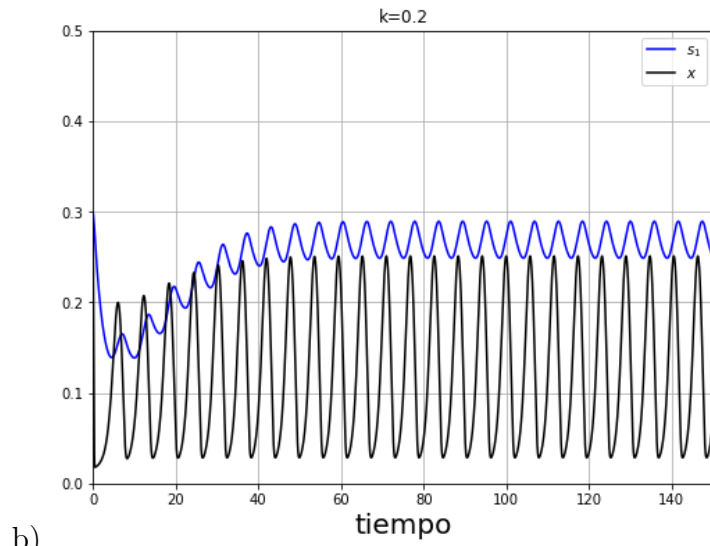
La glucólisis consiste en una serie de reacciones químicas de carácter enzimático, que tienen como objetivo transformar una molécula de glucosa ( $C_6H_{12}O_6$ ) en dos de piruvato ( $C_3H_4O_3$ ) para la adquisición de energía; son dos fases las que completan este proceso, la primera consiste en el gasto de energía al consumir dos unidades de ATP, y la segunda, la obtención de energía, resultando en cuatro moléculas de ATP [3]. Por lo tanto, el acoplamiento O-G, está presente en la obtención de energía celular por la naturaleza de los dos procesos acoplados.

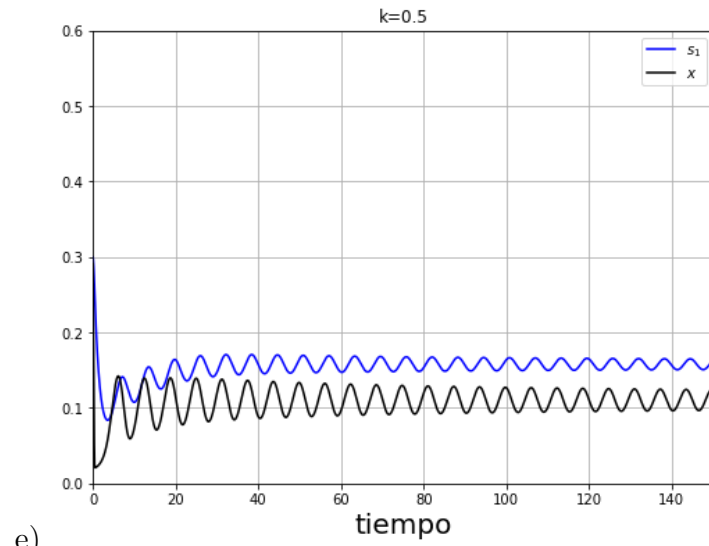
Las soluciones numéricas del acoplamiento O-G, se muestran en las figuras 3.6. Para valores de  $k$  pequeños, el acoplamiento es efectivo y cuando  $k$  toma valores mayores a 0.5 (inciso *e*) las oscilaciones comienzan a amortiguarse. Este tipo de acoplamiento presenta oscilaciones prácticamente al mismo periodo desde  $k=0.1$ .

Por otro lado, es importante notar que aunque la dinámica de glucólisis es muy lenta,  $\sigma_1$  logra oscilar al mismo periodo que  $x$  para valores de  $k = 0.1, 0.2, 0.3$  y  $0.4$  (imágenes *a*), *b*), *c*) y *d*)), a la vez que la amplitud de  $\sigma_1$  disminuye casi a la misma amplitud que  $x$ . Los acoplamientos más efectivos se presentan cuando  $k = 0.3$  y  $k = 0.4$ .

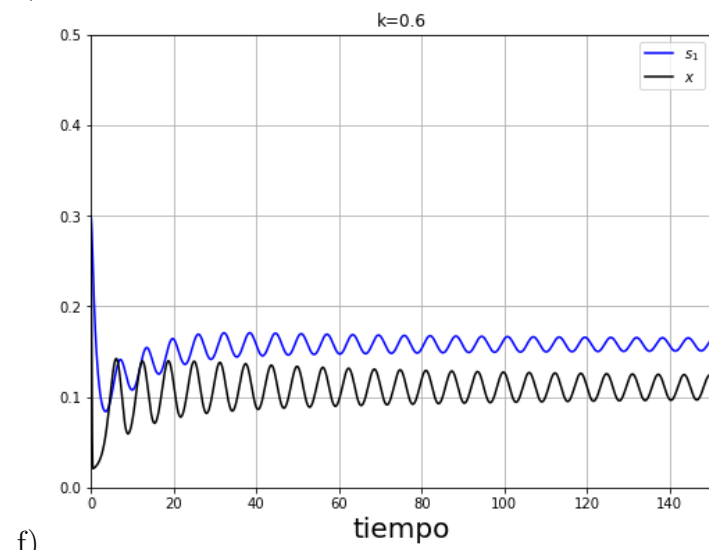
A partir de  $k = 0.5$  el proceso de amortiguamiento, debido a que  $k$  es demasiado abrasivo, comienza a notarse, hasta que el modelo deja de presentar oscilaciones (inciso *g*)). Cuando  $k = 0.5$  las amplitudes de las variables  $x$  y  $\sigma_1$  son prácticamente iguales, hasta que las oscilaciones son atenuadas.



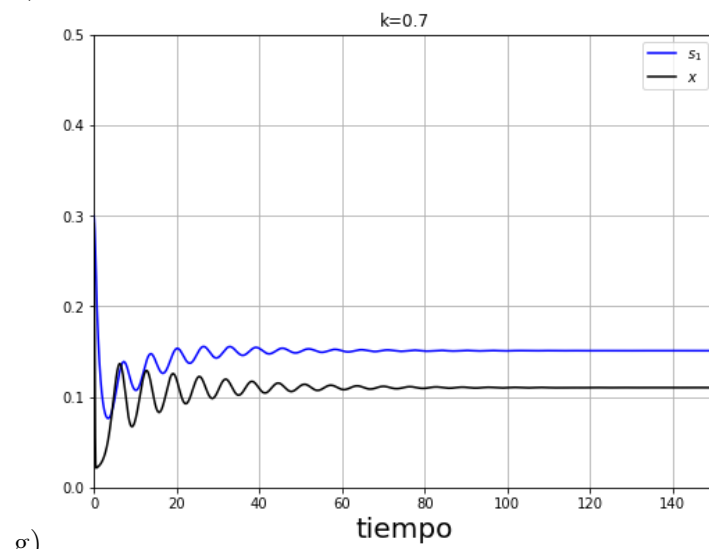




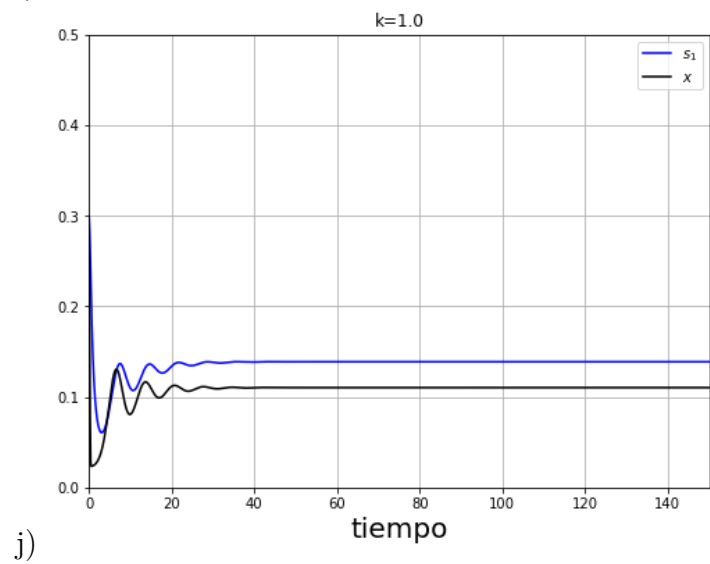
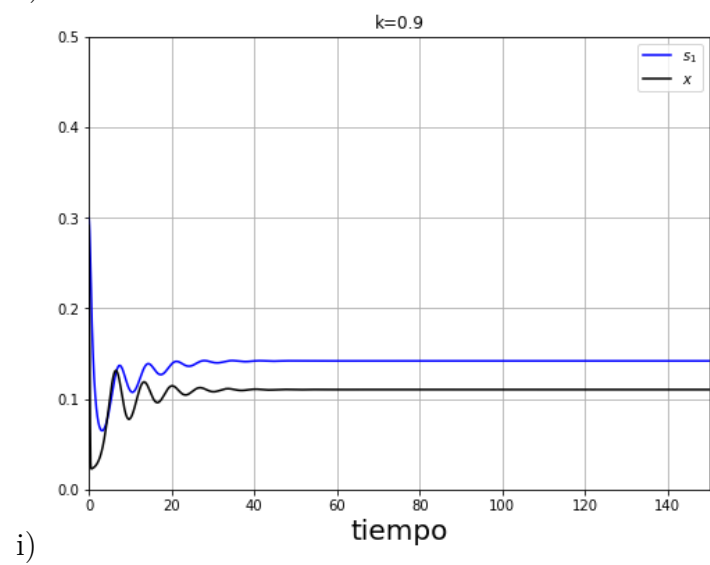
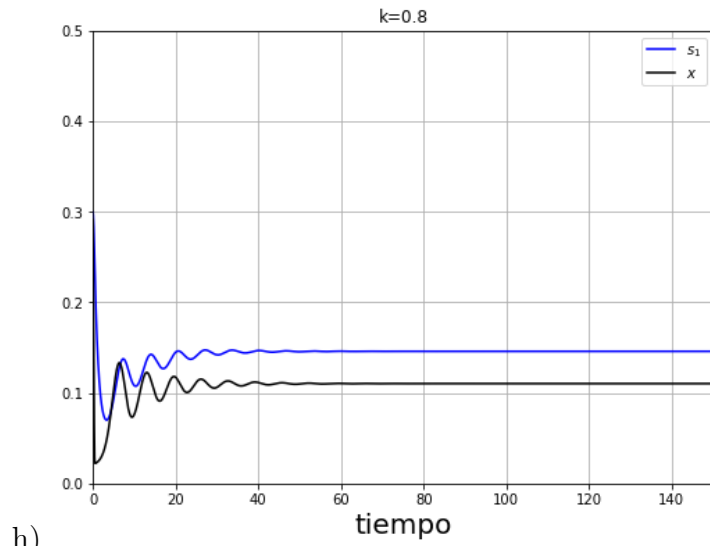
e)

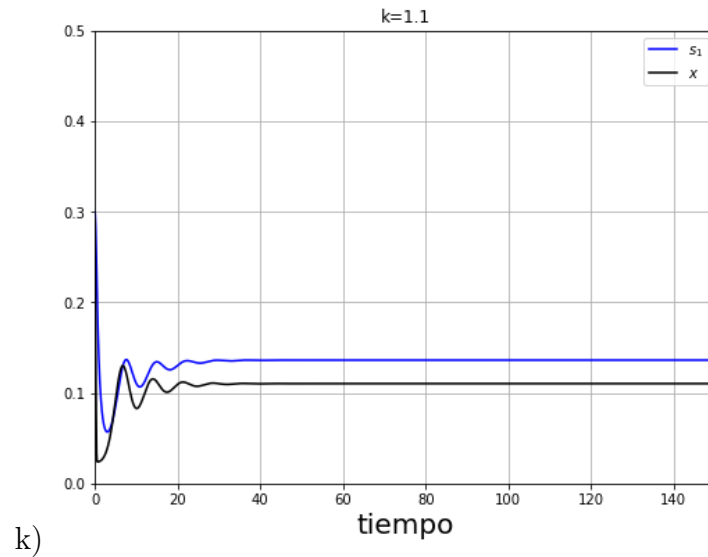


f)



g)





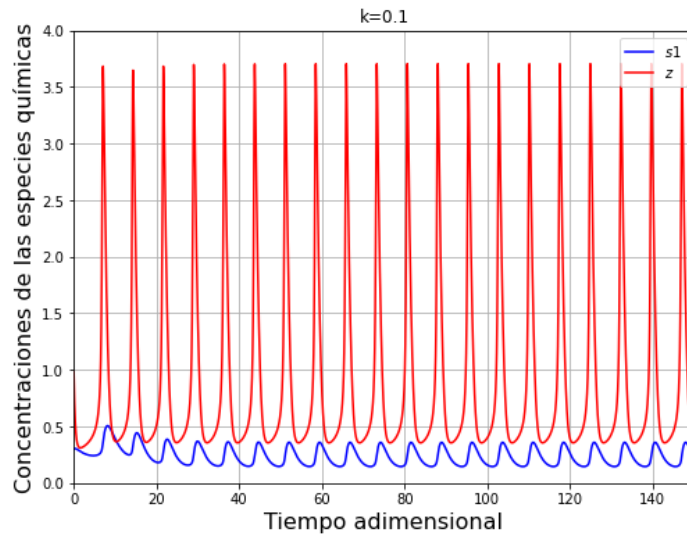
**Figura 3.6:** Solución numérica para el acoplamiento bidireccional lineal de los sistemas oregonator con glucólisis y oscilaciones glicolíticas para valores de  $k = 0.1, 0.2, \dots, 1.1$ . Es de notar que a medida que  $k$  aumenta, las oscilaciones comienzan a desaparecer.

### 3.1.1.3. Glucólisis y oscilaciones glicolíticas con brusselator (G-B).

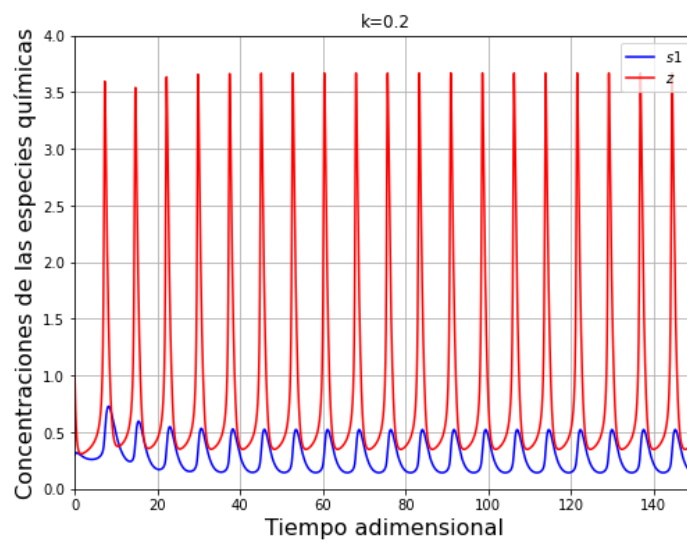
El acoplamiento G-B, al igual que el acoplamiento O-B, está presente en el desarrollo celular, ya que el proceso de glucólisis es una de las vías principales de adquisición de energía para la célula y en el desarrollo de la célula existen procesos trimoleculares como es la formación de ozono ( $O_3$ ) a partir del oxígeno ( $O_2$ ).

Las soluciones de este acoplamiento se muestran en las figuras 3.7. Al igual que en el acoplamiento O-B, la sincronización de  $\sigma_1$  y  $z$  es más efectiva a medida que el parámetro de sincronización  $k$  aumenta, tanto en amplitud como en frecuencia. La amplitud de  $z$  se ve disminuida a medida que  $k$  aumenta, mientras que la amplitud de  $\sigma_1$  aumenta.

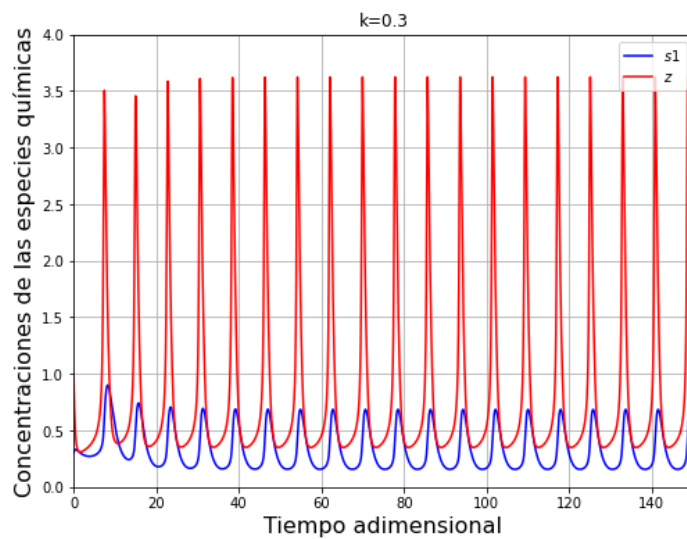
En este acoplamiento no se presenta el fenómeno de amortiguamiento como en O-G, en nuestro rango de estudio, pero para valores  $k$  más grandes, el amortiguamiento está presente.



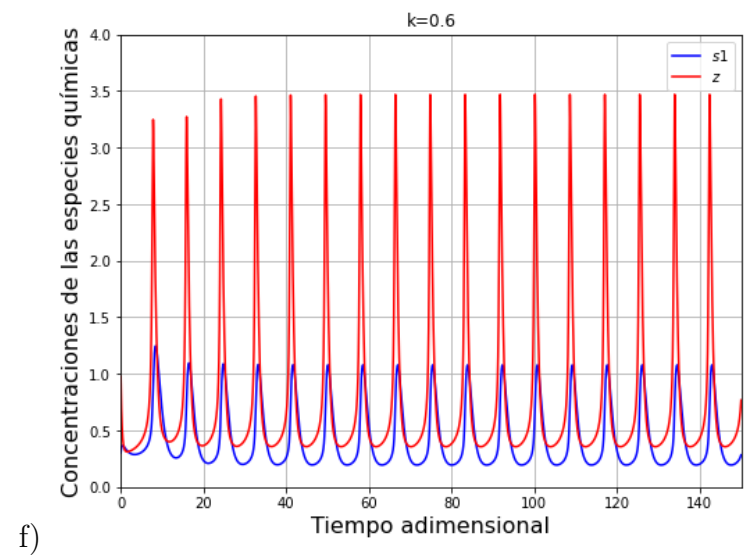
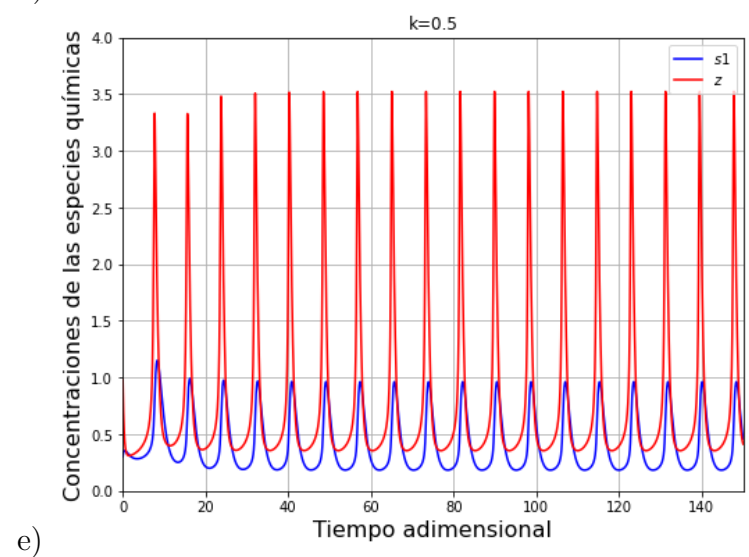
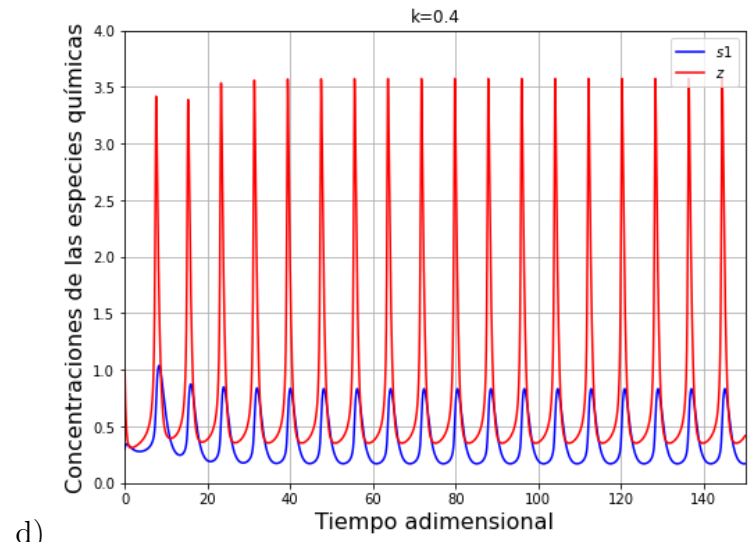
a)

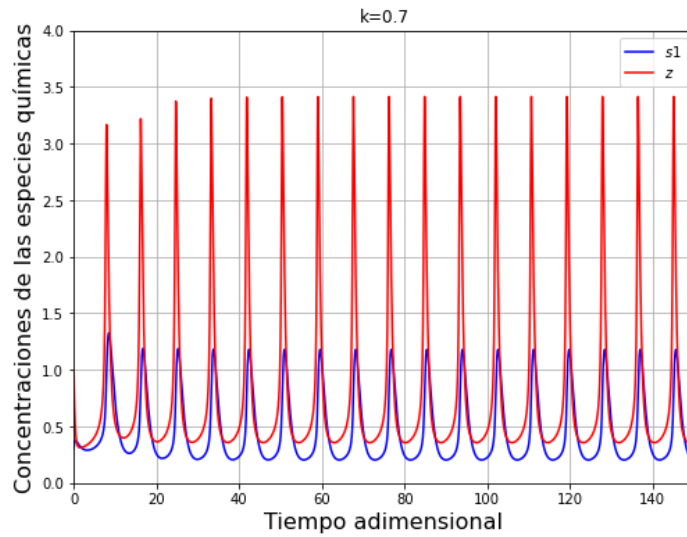


b)

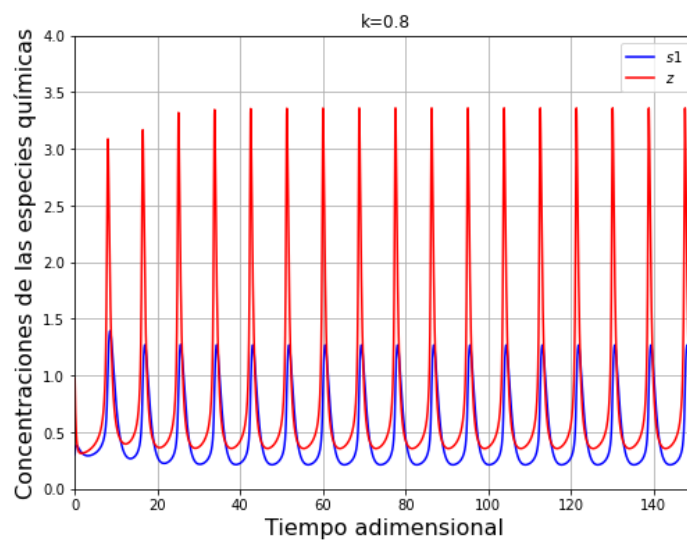


c)

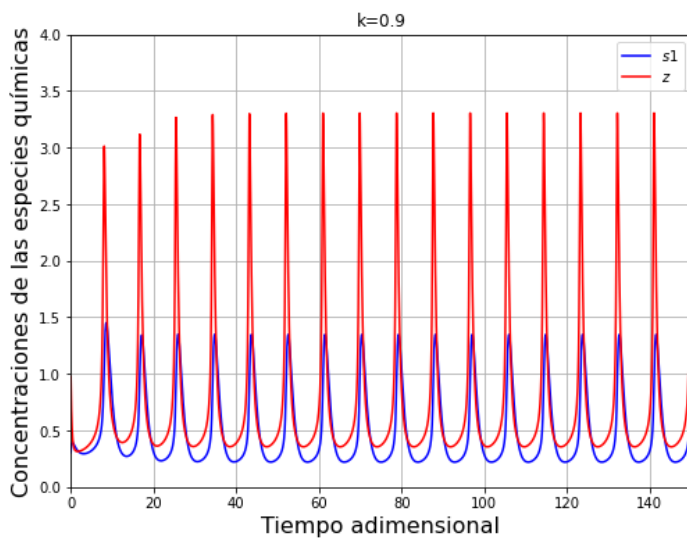




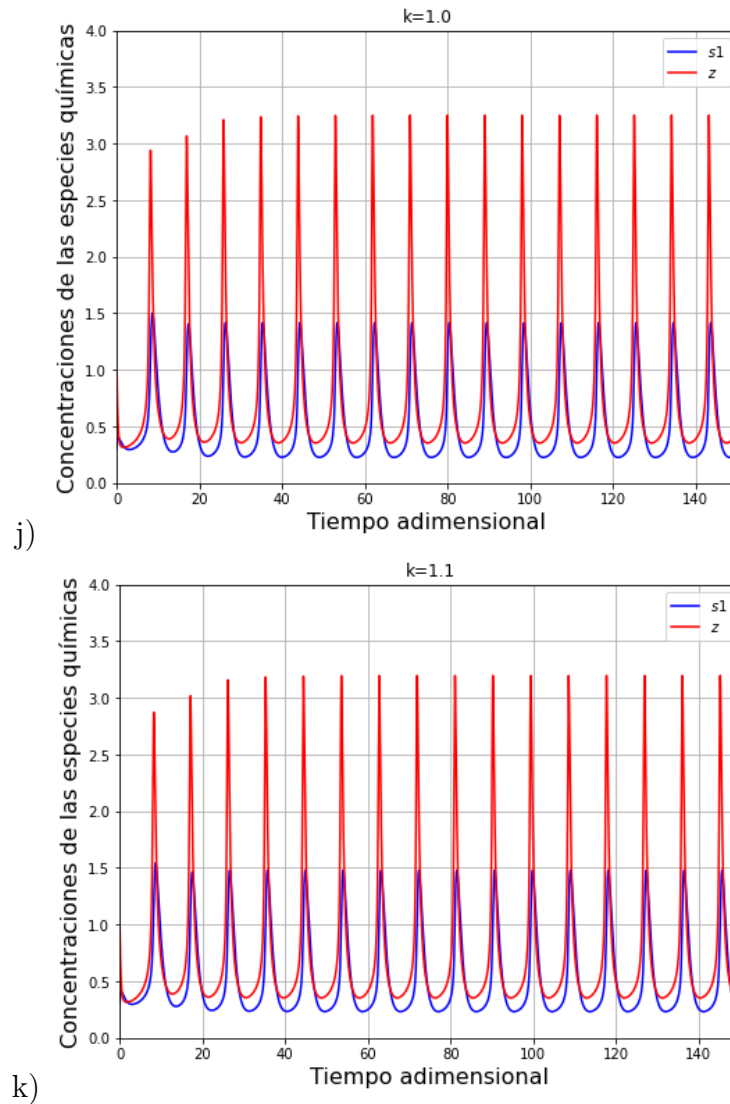
g)



h)



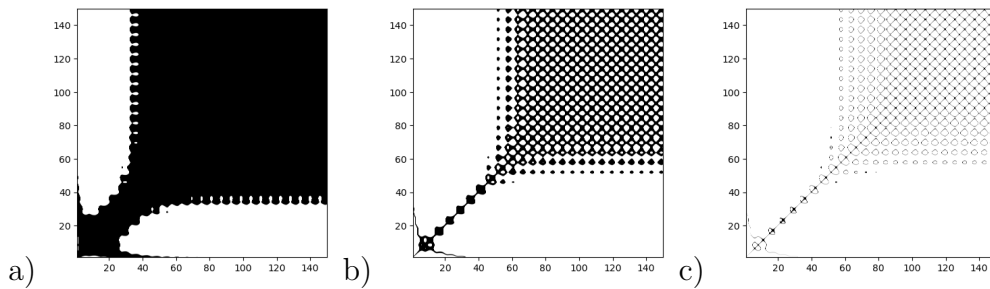
i)



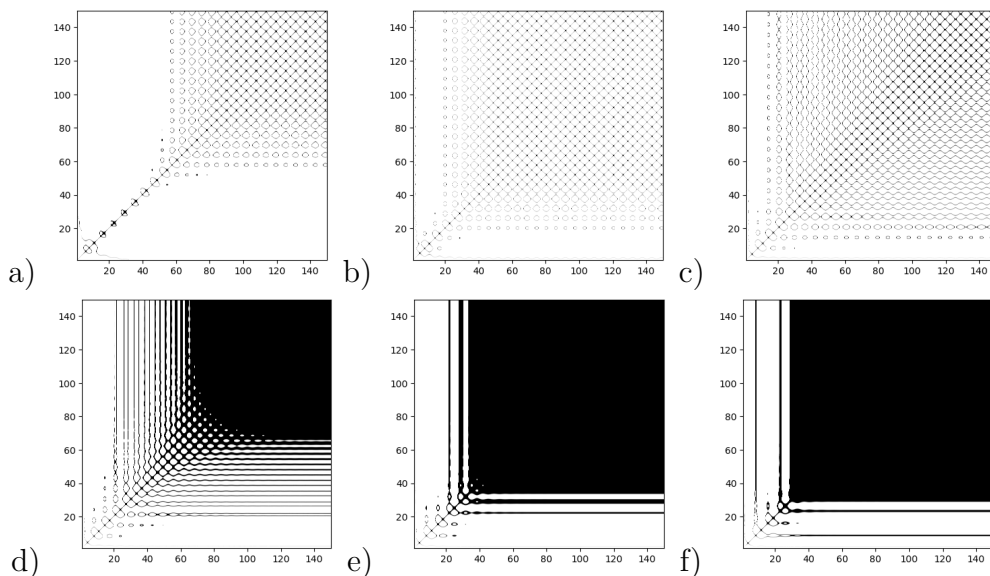
**Figura 3.7:** Solución numérica para el acoplamiento bidireccional lineal de los sistemas brusselator y glucólisis y oscilaciones glicolíticas para valores de  $k = 0.1, 0.2, \dots, 1.1$ .

### 3.2. Gráficas de recurrencia

Después, con los datos obtenidos de la solución numérica de los diferentes acoplamientos, se obtuvieron las gráficas de recurrencia, con un umbral de  $\epsilon = 0.1, 0.01$  y  $0.001$  para cada valor de  $k$  en cada acoplamiento y para cada variable [31]. Así, tenemos 3 diferentes etiquetas de clase con 66 datos de entrada cada clase. A manera de ilustración mostramos unos cuantos diagramas correspondientes a la variable  $\sigma_1$  en el acoplamiento O-G.



**Figura 3.8:** Gráficas de recurrencia para  $\sigma_1$  y  $k=0.1$  en el acoplamiento O-G, con umbrales de 0.1, 0.01 y 0.001 (de izquierda a derecha). Notar que a medida que el umbral se hace más pequeño, los puntos negros disminuyen debido a que el umbral es más estricto. En todas las gráficas de recurrencia la diagonal siempre va a aparecer con un punto negro ya que se está comparando el  $i$ -ésimo valor consigo mismo. Para valores cercanos a este, el cambio en las concentraciones es mínimo por lo que es común, encontrar valores cercanos a la diagonal. Lo que identifica a cada acoplamiento de los demás, son los valores lejanos a esta y los patrones que se forman.



**Figura 3.9:** Gráficas de recurrencia para  $\sigma_1$ , umbral  $\epsilon = 0.001$  y  $k = 0.1, 0.3, 0.5, 0.7, 0.9$  y  $1.1$  (de izquierda a derecha y de arriba hacia abajo). Podemos notar que a medida de  $k$  aumenta, se forma un patrón bien definido y representativo del acoplamiento. Para valores de acoplamiento efectivo, se obtienen puntos negros cercanos a la diagonal. Como estamos en el acoplamiento O-G, el amortiguamiento (incisos d) e) y f)) se muestra en un continuo de puntos negros en la sección superior izquierda.

Las gráficas de recurrencia revelan patrones bien definidos y únicos para cada aco-

plamiento, simulando un grado de sincronía proporcional al valor del parámetro de acoplamiento, formando así un conjunto de datos suficiente para poder entrenar un algoritmo de aprendizaje supervisado y este nos permita clasificar cada diagrama en su tipo de acoplamiento.

### 3.3. Implementación de los modelos de aprendizaje automático supervisado

Para la implementación de los algoritmos de aprendizaje supervisado se empleó el software Orange 3, el cual, es un entorno gráfico con componentes desarrollados en C++ que implementa algoritmos especializados en dos grandes ramas de la estadística moderna: *la minería de datos y el procesamiento de datos para análisis predictivo*.

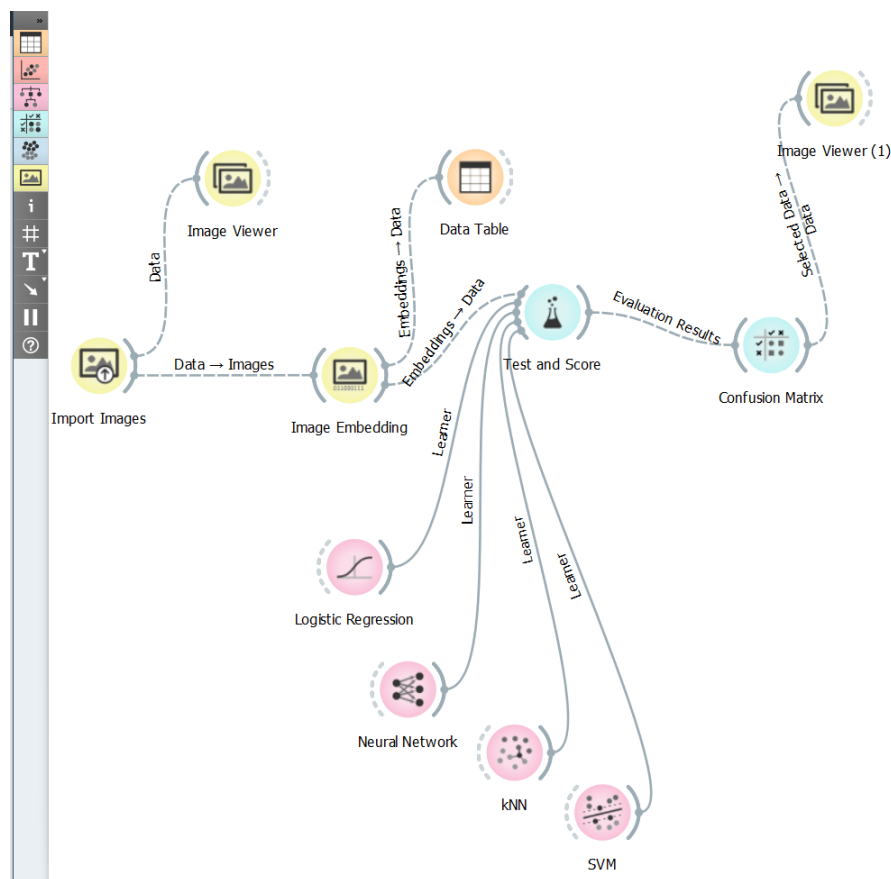
La minería de datos es un campo de la estadística y las ciencias de la computación que tiene como objetivo extraer información de un conjunto de datos y transformarla en una estructura comprensible para su uso posterior. En cambio, el análisis predictivo agrupa una variedad de técnicas estadísticas de modelación y aprendizaje automático analizando los datos actuales y pasados para hacer predicciones acerca del futuro o acontecimientos no conocidos [8]. Los modelos de aprendizaje automático, son aquellos nombrados en la sección 2.6, los cuales corresponden a modelos de clasificación. Los componentes en Orange son denominados *widgets*. Los widgets son herramientas que permiten funcionalidades básicas como leer los datos, seleccionar funciones, entrenar predictores, comparar algoritmos de aprendizaje, visualizar elementos de datos, etcétera y son organizados en la interfaz a manera de creación de un flujo de trabajo, denominado *workflow*. De esta forma, el conjunto de datos de entrada pasa por diferentes procesos para obtener la salida deseada [9].

El algoritmo de aprendizaje automático implementado se muestra en la figura 3.10, este, está conformado por tres tipos de widgets: procesamiento de datos, modelos predictivos y evaluación de resultados [9].

#### 3.3.0.1. Widgets de gestión de datos

Son aquellos de color amarillo en la figura 3.10, tienen como objetivo recolectar, convertir y adecuar los datos de entrada, en este caso, los datos son obtenidos de las gráficas de recurrencia. [9].

**Import Images:** Este widget recorre un directorio y devuelve una fila por imagen ubicada. Las columnas incluyen el nombre de la imagen, la ruta a la imagen, el ancho, la altura y el tamaño de la imagen. Es posible cargar una carpeta que contenga varias subcarpetas. En este caso, Orange considerará cada subcarpeta como



**Figura 3.10:** Modelo propuesto para la clasificación del acoplamiento bidireccional lineal de sistemas dinámicos.

un valor de clase. Así, las imágenes de las gráficas de recurrencia se encuentran en tres subcarpetas correspondientes a cada acoplamiento, representando cada carpeta un valor de clase (O-B, O-G y G-B).

**Image Viewer:** Permite mostrar las imágenes previamente importadas, en este caso, el conjunto de gráficas de recurrencia.

**Image Embedding:** El objetivo de este widget es leer imágenes y cargarlas en un servidor remoto o local. Estos servidores contienen modelos de aprendizaje profundo (redes neuronales convolucionales previamente entrenadas) y son utilizados para calcular un vector de características para cada imagen, de esta manera, las imágenes cargadas en el servidor, son devueltas como tablas de datos mejoradas con columnas adicionales conocidas como *descriptores de imagen*, a este proceso, se le denomina *aprendizaje por transferencia*. El aprendizaje por transferencia consiste en aplicar el conocimiento que hayamos obtenido en un dominio particular para resolver problemas en otro dominio, generalmente relacionado con el primero. Este tipo de aprendizaje resulta demasiado útil si no se dispone de demasiados datos etiquetados para la resolución del problema particular.

Para problemas de visión artificial y etiquetado de datos, como es el caso, las *Redes Neuronales Convolutivas (CNN)* suelen obtener resultados excelentes.

Una CNN, en general, consiste en un conjunto de unidades, llamadas neuronas artificiales, conectadas entre sí para transmitirse señales. La información de entrada atraviesa la red neuronal (donde se somete a diversas operaciones) produciendo unos valores de salida. Cada neurona está conectada con otras a través de unos enlaces. En estos enlaces el valor de salida de la neurona anterior es multiplicado por un valor de peso. De esta manera, las entradas van cambiando su valor al pasar entre cada neurona hasta encontrar la salida correspondiente para la cual la red está construida; en procesamiento digital, las entradas son matrices, y las funciones de peso realizan multiplicaciones matriciales comunes de las entradas por pesos.

Las CNN son redes neuronales artificiales comunes solo que en algunos enlaces se realiza una operación de convolución en lugar de la multiplicación por pesos [47]. Matemáticamente, la convolución es una operación que se realiza sobre dos funciones para producir una tercera que se suele interpretar como una versión simplificada (filtrada) de una de las funciones originales, y es definida como:

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m] = \sum_{m=-\infty}^{\infty} f[n-m]g[m] \quad (3.5)$$

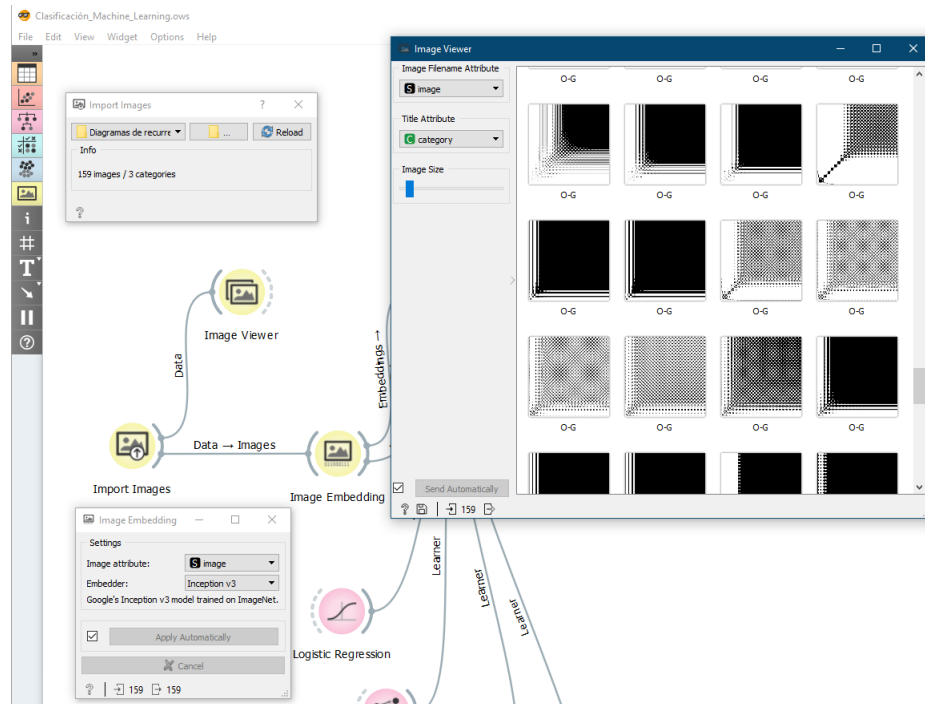
claramente esta operación se puede extender al caso bidimensional, uno de los operandos de la convolución es la señal que deseamos procesar:  $x[x, y]$ , donde  $[x, y]$  corresponden a las coordenadas de los píxeles de cada gráfica, el otro operando, corresponde al filtro representado como  $h[k_1, k_2]$ , que es con el que procesamos la señal, este se define sólo sobre el dominio  $0, 1, \dots, K_1 - 1$  y  $0, 1, \dots, K_2 - 1$  y tamaño  $K_1 \times K_2$ , la operación de convolución consiste en, realizar  $K_i$  multiplicaciones y  $K_i - 1$  sumas para cada valor de la señal, de la siguiente manera:

$$(x * h)[x, y] = \sum_{k_1=0}^{K_1-1} \sum_{k_2=0}^{K_2-1} h[k_1, k_2]x[x+k_1, y+k_2] \quad (3.6)$$

Las CNN's utilizadas en este trabajo son:

- *Inception V3*: Desarrollada por Google, esta CNN de reconocimiento de imágenes fue desarrollada para el análisis de imágenes y detección de objetos. El modelo fue entrenado con el conjunto de datos de ImageNet obteniendo una exactitud superior al 78.1%. El modelo está formado por 48 capas que incluyen operaciones de convolución, reducción promedio, reducción máxima y concatenaciones, entre otras operaciones. La normalización por lotes se usa con frecuencia en todo el modelo y se aplica a las entradas de activación. Para la vectorización de las imágenes usamos la activación de la penúltima capa del modelo, cada imagen es representada con un vector de dimensión 2048. En este trabajo de tesis empleamos estos vectores como descriptores de cada imagen [43].

- *SqueezeNet*: esta CNN fue desarrollada por investigadores de DeepScale, la Universidad de California, Berkeley y la Universidad de Stanford. Esta CNN también se emplea para la clasificación de imágenes y detección de objetos, el modelo logra una precisión al mismo nivel que la CNN AlexNet en la base de datos de ImageNet (15.3% de error) con 50 veces menos parámetros. La red SqueezeNet representa a cada imagen con vectores de 1000 entradas [20].



**Figura 3.11:** Interfaces gráficas de cada widget de gestión. La ventana del widget abre un directorio para importar la carpeta que contiene las variables de clase, mientras que el widget de Image Embedding nos permite seleccionar la red CNN para calcular los vectores con los descriptores de imagen. Por último el widget Image Viewer simplemente nos permite navegar entre las gráficas de los diagramas de recurrencia.

#### 3.3.0.2. Widgets de modelos predictivos

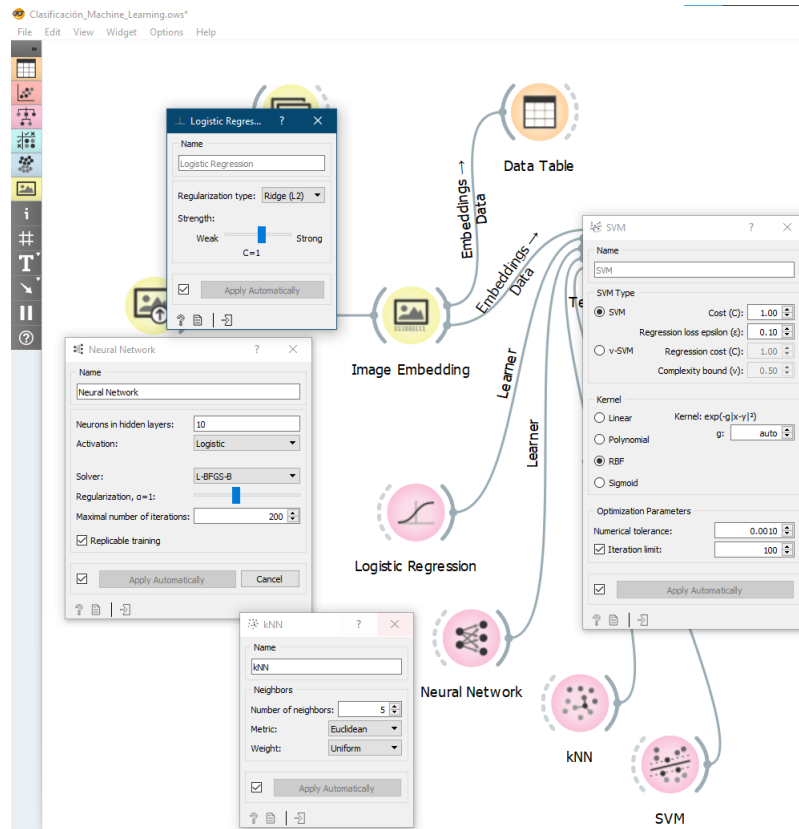
El objetivo de los widgets color rosa en la figura 3.10 tienen como objetivo realizar la clasificación de cada gráfica de recurrencia en un valor de clase. Cada widget representa un algoritmo de clasificación.

**Logistic Regression:** Este widget aprende un modelo de *regresión logística* a partir del conjunto de datos.

**Neural Network:** El widget utiliza el algoritmo de red neuronal perceptrón multicapa.

**kNN:** El widget kNN usa el algoritmo kNN (k-nearest neighbors).

**SVM:** Este widget aprende el algoritmo de las máquinas de soporte vectorial (support vector machines).

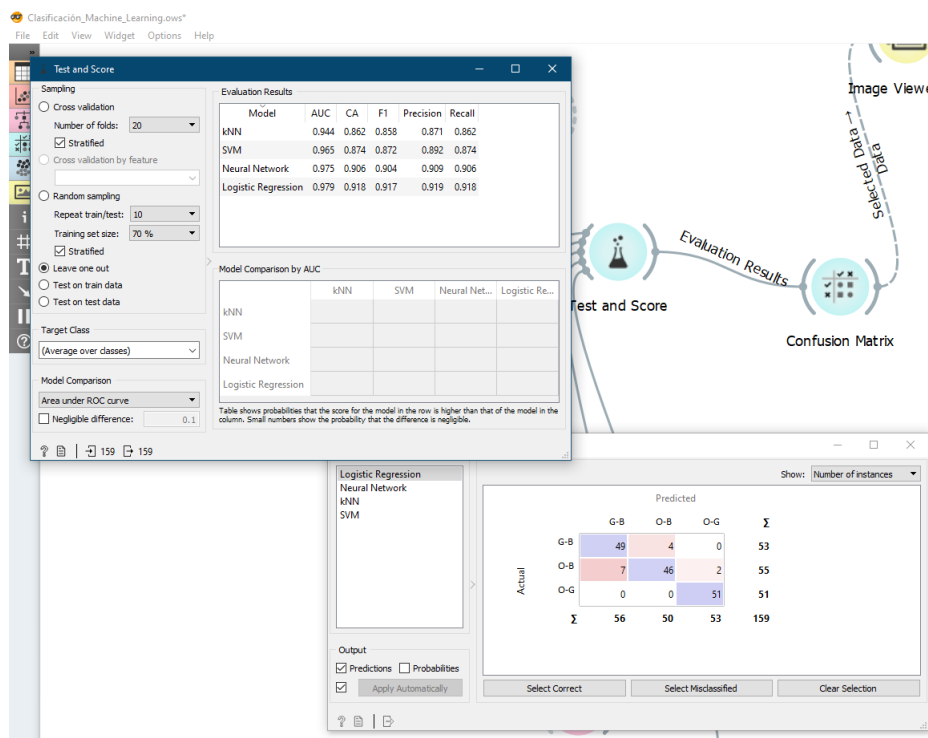


**Figura 3.12:** Interfaces gráficas de cada widget de modelos predictivos. En el modelo Logistic Regression el tipo de regularización L2 utiliza la función de verosimilitud (ecuación 2.35). El widget Neural Network nos permite seleccionar el número de neuronas en una capa oculta en el perceptrón multicapa (en este caso 10) con funciones de activación logísticas; el solucionador L-BFGS-B utiliza la ecuación 2.37 y  $\alpha$  es un parámetro asociado a la regularización L2. El widget que aprende un modelo kNN nos permite seleccionar el número de vecinos para realizar la predicción, el tipo de métrica y el peso que tengan cada uno de estos vecinos. Por último, los parámetros del algoritmo SVM, son la penalización  $C = 1/L2$  y  $\xi_i$ , la pérdida en la regresión de la ecuación 2.39, con la función del núcleo nombrada en la ecuación 2.40.

### 3.3.0.3. Widgets de evaluación del rendimiento

**Test and Score:** El widget prueba los algoritmos de predicción. Los dos métodos para evaluar los modelos predictivos son: *validación cruzada estratificada* y *leave one out*.

**Confusion Matrix:** Este widget genera la matriz de confusión para cada modelo predictivo.



**Figura 3.13:** Interfaces gráficas de cada widget de evaluación. El widget Test and Score nos permite seleccionar el tipo de entrenamiento y evaluación (en el caso de la validación cruzada, es posible seleccionar el número de pliegues y si esta será estratificada o no). Por otro lado, el widget Confusion Matrix muestra el número de instancias y cada una de las predicciones de cada modelo.



## 4 Resultados y discusión

Se obtuvieron las gráficas de recurrencia de cada acoplamiento, con un umbral de  $\epsilon = 0.1, 0.01$  y  $0.001$  para valores de  $k = 0.1, 0.2, \dots, 1.1$  en cada una de las dos variables acopladas. Resultando así 66 gráficas de recurrencia por cada clase. Al calcular las gráficas para  $\epsilon = 0.1$ , los patrones de recurrencia no eran bien definidos por lo que se decidió eliminar varias de las imágenes que no contribuían al entrenamiento de los modelos. Obteniendo como resultado 159 gráficas de recurrencia distribuidas entre cada clase como se indica en el cuadro 4.1.

**Cuadro 4.1:** Número de diagramas por clase.

Valor de clase	Número de diagramas
O-B	55
O-G	51
G-B	53

Se realizó la vectorización de las 159 gráficas de recurrencia mediante la implementación de dos CNN, Inception V3 y SqueezeNet. Después empleando como métodos de entrenamiento la validación cruzada estratificada de 20 pliegues y el método leave one out, se entrenaron los cuatro modelos de aprendizaje automático descritos en el apartado 2.6.1: regresión logística, red neuronal, kNN y SVM). Las métricas de evaluación empleadas (sección 2.6.3), son las siguientes: AUC, CA, F1, Precisión y Recall.

Resumiendo las métricas, la prueba CA nos sirve para conocer qué tan correctas son nuestras clasificaciones afirmativas tanto positivas como negativas. La precisión se utiliza como métrica de rendimiento cuando el objetivo es limitar el número de falsos positivos. La medida recall se utiliza como métrica de rendimiento cuando necesitamos identificar todas las muestras positivas. El puntaje F1 corresponde a un resumen entre la precisión y el recall. Por último, AUC representa el grado o medida de separabilidad, es decir, representa cuánto es capaz el modelo de distinguir entre clases.

## 4.1. Inception V3

### 4.1.1. Validación cruzada estratificada

Por el método de validación cruzada estratificada con 20 pliegues se obtienen los resultados enlistados en el cuadro 4.2, el cual, corresponde a las matrices de confusión de cada modelo de aprendizaje supervisado.

**Cuadro 4.2:** Matrices de confusión para cada modelo predictivo por el método de validación cruzada estratificada de 20 pliegues de la red Inception V3.

Regresión logística:

		Predicha			$\Sigma$
		G-B	O-B	O-G	
Real	G-B	50	3	0	53
	O-B	8	45	2	55
	O-G	1	0	50	51
	$\Sigma$	59	48	52	159

Red neuronal:

		Predicha			$\Sigma$
		G-B	O-B	O-G	
Real	G-B	50	3	0	53
	O-B	8	43	4	55
	O-G	1	0	50	51
	$\Sigma$	59	48	52	159

kNN:

		Predicha			$\Sigma$
		G-B	O-B	O-G	
Real	G-B	49	2	2	53
	O-B	14	38	3	55
	O-G	0	1	50	51
	$\Sigma$	59	48	52	159

SVM:

		Predicha			$\Sigma$
		G-B	O-B	O-G	
Real	G-B	52	1	0	53
	O-B	14	39	2	55
	O-G	2	1	48	51
$\Sigma$		59	48	52	159

En la clasificación de las gráficas de recurrencia de la clase G-B (cuadro 4.3; todos los cuadros pertenecientes a la clase G-B aparecerán en letras azules para facilitar su interpretación), los 4 modelos de aprendizaje presentan un puntaje sobresaliente en las métricas AUC y recall, por lo cual, podemos caracterizar al acoplamiento G-B por su separabilidad de los otros dos, ya que el AUC precisamente mide el grado de distinción entre clases, mientras que el recall mide todas las muestras positivas. Por otro lado, los puntajes más bajos, se dan en la precisión, de esta manera, este acoplamiento tiende a recibir falsos positivos.

**Cuadro 4.3:** Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase G-B por el método de validación cruzada estratificada de 20 pliegues.

Modelo	AUC	CA	F1	Precisión	Recall
kNN	0.945	0.887	0.845	0.778	0.925
SVM	0.964	0.893	0.860	0.765	0.981
Red Neuronal	0.965	0.925	0.893	0.847	0.943
Regresión Logística	0.967	0.925	0.893	0.847	0.943

El cuadro 4.4 representa los resultados de la clasificación del acoplamiento O-B (cuadros con letras rojas). Podemos notar que el modelo kNN disminuye considerablemente su desempeño, también es notoria una disminución considerable en la métrica recall y un aumento en la precisión.

**Cuadro 4.4:** Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase O-B por el método de validación cruzada de 20 pliegues.

Modelo	AUC	CA	F1	Precisión	Recall
kNN	0.904	0.874	0.792	0.927	0.691
SVM	0.953	0.887	0.812	0.951	0.709
Red Neuronal	0.953	0.906	0.851	0.935	0.782
Regresión Logística	0.964	0.918	0.874	0.938	0.818

Al clasificar la clase O-G (cuadro 4.5, de igual manera que en las otras clases, los cuadros correspondientes a esta clase aparecerán con letras de color verde azulado),

los modelos de aprendizaje obtienen los mejores resultados de las tres clases. Siendo un acoplamiento con un grado de separabilidad ideal permitiendo a todos los modelos obtener una puntuación muy cercana a 1.0 en AUC, también son destacables sus resultados en CA la cual mide qué tan correctas son las clasificaciones tanto positivas como negativas.

**Cuadro 4.5:** Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase O-G por el método de validación cruzada estratificada de 20 pliegues.

Modelo	AUC	CA	F1	Precisión	Recall
kNN	0.996	0.962	0.943	0.909	0.980
SVM	0.997	0.969	0.950	0.960	0.941
Red Neuronal	1.0	0.969	0.952	0.926	0.980
Regresión Logística	1.0	0.981	0.971	0.962	0.980

El objetivo de este estudio es generalizar la metodología para la clasificación de estados en procesos fisiológicos sincronizados. Al promediar los resultados anteriores, obtenemos la tabla de resultados AOC (cuadro 4.6). Los modelos destacan por sus buenos resultados en la métrica AUC, de esta manera, los modelos predictivos son capaces de tener una buena separabilidad al clasificar cada acoplamiento.

**Cuadro 4.6:** Tabla de resultados con el promedio de los resultados AOC en cada clase por el método de validación cruzada estratificada de 20 pliegues.

Modelo	AUC	CA	F1	Precisión	Recall
kNN	0.946	0.862	0.858	0.871	0.862
SVM	0.968	0.874	0.872	0.892	0.874
Red Neuronal	0.977	0.899	0.898	0.903	0.899
Regresión Logística	0.981	0.912	0.911	0.915	0.912

El modelo que obtuvo los resultados más altos, al implementar esta red y para este tipo de entrenamiento, es el de regresión logística, obteniendo los resultados más altos en todas y cada una de las métricas. Por último, la matriz Bayessiana de esta CNN entrenada por el método de validación cruzada (cuadro 4.7), corrobora los resultados obtenidos por la regresión logística. En general, los resultados del modelo kNN en las métricas no son bajos, pero en este caso, cualquier otro modelo de aprendizaje tiene una probabilidad más alta de obtener resultados más altos. En conclusión, para esta red y este tipo de aprendizaje, el modelo de aprendizaje supervisado con mejor desempeño es la regresión logística.

**Cuadro 4.7:** Matriz Bayessiana con el análisis entre modelos para el método de validación cruzada de 20 pliegues de la red Inception V3.

	kNN	SVM	Red neuronal	Regresión logística
kNN		0.068	0.034	0.024
SVM	.932		0.405	0.118
Red neuronal	0.966	0.595		0.185
Regresión logística	0.976	0.882	0.815	

### 4.1.2. Leave One Out

Las matrices de confusión correspondientes a cada modelo predictivo entrenado por el método leave one out son enlistadas en el cuadro 4.8.

**Cuadro 4.8:** Matrices de confusión para cada modelo predictivo por el método leave one out de la red Inception V3.

Regresión logística:

		Predicha			$\Sigma$
		G-B	O-B	O-G	
Real	G-B	49	4	0	53
	O-B	7	46	2	55
	O-G	0	0	51	51
	$\Sigma$	59	48	52	159

Red neuronal:

		Predicha			$\Sigma$
		G-B	O-B	O-G	
Real	G-B	50	3	0	53
	O-B	8	44	3	55
	O-G	1	0	50	51
	$\Sigma$	59	48	52	159

kNN:

		Predicha			$\Sigma$
		G-B	O-B	O-G	
Real	G-B	49	2	2	53
	O-B	14	38	3	55
	O-G	0	1	50	51
$\Sigma$		59	48	52	159

SVM:

		Predicha			$\Sigma$
		G-B	O-B	O-G	
Real	G-B	52	1	0	53
	O-B	14	39	2	55
	O-G	2	1	48	51
$\Sigma$		59	48	52	159

Ahora, si nos ponemos a analizar las matrices de confusión, en general (en ambos métodos de aprendizaje cuadros 4.2 y 4.8), existe una tendencia. Cuando los modelos predictivos clasifican la clase O-B, estos tienden a confundirla con la clase G-B. Esto debido a que al acoplar los sistemas, el modelo brusselator suele registrar la dinámica del acoplamiento.

En general los resultados de la clasificación de la clase G-B (cuadro 4.9), comparados con la validación cruzada (cuadro 4.3) son muy similares, aunque el método de entrenamiento leave one out obtiene resultados más fiables por su carácter exhaustivo. En este cuadro, podemos notar que nuevamente la regresión logística es la que obtiene mejores resultados en tres de las cuatro métricas, solo que en este caso, el modelo SVM obtiene el mayor puntaje en recall.

**Cuadro 4.9:** Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase G-B por el método de leave one out.

Modelo	CA	F1	Precisión	Recall
kNN	0.887	0.845	0.778	0.925
SVM	0.893	0.860	0.765	0.981
Red neuronal	0.925	0.893	0.847	0.943
Regresión logística	0.931	0.899	0.875	0.925

Por otro lado, al comparar los cuadros 4.10 y el 4.4, correspondientes a la clase O-B, los resultados para cada entrenamiento, son muy similares, por lo que este acoplamiento también se caracteriza por obtener pocos falsos positivos.

**Cuadro 4.10:** Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase O-B por el método de leave one out.

Modelo	CA	F1	Precisión	Recall
kNN	0.874	0.792	0.927	0.691
SVM	0.887	0.812	0.951	0.709
Red neuronal	0.912	0.863	0.936	0.800
Regresión logística	0.918	0.876	0.920	0.836

Por último, el método leave one out en la clase O-G (cuadro 4.11), obtiene resultados mejores o iguales a la validación cruzada estratificada (cuadro 4.5). Los resultados más altos los encontramos en el recall y los más bajos en la precisión, tal como en el anterior entrenamiento.

**Cuadro 4.11:** Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase O-G por el método de leave one out.

Modelo	CA	F1	Precisión	Recall
kNN	0.962	0.943	0.909	0.980
SVM	0.969	0.950	0.960	0.941
Red neuronal	0.975	0.962	0.943	0.980
Regresión logística	0.987	0.981	0.962	1.0

Finalmente, al realizar el estudio AOC (cuadro 4.12), los resultados más altos, siguen siendo al implementar el modelo de regresión logística obteniendo, los puntajes más altos en todas las métricas y una pequeña mejoría que en el entrenamiento pasado. Sin embargo, los puntajes obtenidos en los otros modelos son comparables con las métricas de evaluación obtenidas con el previo entrenamiento.

**Cuadro 4.12:** Tabla de resultados con el promedio de los resultados AOC en cada clase por el método de leave one out.

Modelo	AUC	CA	F1	Precisión	Recall
kNN	0.944	0.862	0.858	0.871	0.862
SVM	0.965	0.874	0.872	0.892	0.874
Red neuronal	0.975	0.906	0.904	0.909	0.906
Regresión logística	0.979	0.918	0.917	0.919	0.918

Los puntajes AOC en la CNN Inception V3 de validación cruzada estratificada y leave one out son muy similares. Primeramente, el número de casillas con puntaje mayor en validación cruzada que en leave one out fue 4, con puntajes iguales 8 y por último, 8 donde leave one out fue mejor que la validación cruzada. La mayor

diferencia en un puntaje se da en el modelo Neural Network en la métrica CA y es de .007 puntos para leave one out mejor que la validación cruzada. Esta diferencia de puntaje no es determinante para conocer qué entrenamiento es mejor que el otro. En teoría, el entrenamiento leave one out debería de ser mejor dado que los modelos son entrenados con un mayor número de datos, pero como ambos procesos obtuvieron puntajes muy similares y cercanos a 1, elegiremos la validación cruzada estratificada de 20 pliegues como método de entrenamiento y evaluación para estudios posteriores, dado que el proceso de leave one out tarda aproximadamente 15 veces más el tiempo de entrenamiento y clasificación, que la validación cruzada estratificada.

## 4.2. SqueezeNet

### 4.2.1. Validación cruzada estratificada

Las matrices de confusión correspondientes a cada modelo predictivo entrenados por el método de validación cruzada, se presentan en el cuadro 4.13. Podemos notar que el número de errores en todos los modelos aumentan al utilizar esta red.

**Cuadro 4.13:** Matrices de confusión para cada modelo predictivo por el método de validación cruzada de 20 pliegues de la red SqueezeNet.

Regresión logística:

		Predicha			$\Sigma$
		G-B	O-B	O-G	
Real	G-B	40	12	1	53
	O-B	6	45	4	55
	O-G	0	3	48	51
	$\Sigma$	59	48	52	159

Red neuronal:

		Predicha			$\Sigma$
		G-B	O-B	O-G	
Real	G-B	43	9	1	53
	O-B	6	44	5	55
	O-G	1	3	47	51
	$\Sigma$	59	48	52	159

kNN:

		Predicha			$\Sigma$
		G-B	O-B	O-G	
Real	G-B	43	8	2	53
	O-B	11	35	9	55
	O-G	1	6	44	51
	$\Sigma$	59	48	52	159

SVM:

		Predicha			$\Sigma$
		G-B	O-B	O-G	
Real	G-B	43	10	0	53
	O-B	9	40	6	55
	O-G	2	3	46	51
	$\Sigma$	59	48	52	159

Por el método de validación cruzada estratificada con 20 pliegues se obtiene para cada modelo los siguientes resultados en las métricas.

El cuadro de resultados de la clase G-B (cuadro 4.14) revela que el recall comparado con la otra red (cuadros 4.3 y 4.9), disminuye considerablemente (en más de 0.1 puntos en todos los modelos predictivos) y por lo tanto también disminuye la métrica F1. El puntaje F1 nos da un equilibrio entre la precisión y el recall, es decir, nos da una estimación del número de falsos positivos en el total de positivos, contrastando la precisión y el recall. Sabiendo que una puntuación baja en la medida de precisión corresponde a obtener muchos falsos positivos, si el recall también es bajo, estas puntuaciones corresponden a tener pocos verdaderos positivos.

**Cuadro 4.14:** Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase G-B por el método de validación cruzada estratificada de 20 pliegues.

Modelo	AUC	CA	F1	Precisión	Recall
kNN	0.927	0.862	0.796	0.782	0.811
SVM	0.952	0.868	0.804	0.796	0.811
Red neuronal	0.963	0.893	0.835	0.860	0.811
Regresión logística	0.956	0.881	0.808	0.870	0.755

En la clasificación de la clase O-B, (el cuadro 4.15) nos permite ver que las métricas precisión y F1 también se ven disminuidas.

**Cuadro 4.15:** Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase O-B por el método de validación cruzada estratificada de 20 pliegues.

Modelo	AUC	CA	F1	Precisión	Recall
kNN	0.816	0.786	0.673	0.714	0.636
SVM	0.903	0.824	0.741	0.755	0.727
Red neuronal	0.927	0.855	0.793	0.786	0.800
Regresión logística	0.935	0.843	0.783	0.750	0.818

En los resultados de la clase O-G (cuadro 4.16), nuevamente observamos una disminución en el puntaje de prácticamente todas las métricas, pero es de resaltar, su resultado en AUC. Este acoplamiento al obtener buenos puntajes en AUC se caracteriza por su separabilidad de los otros dos.

**Cuadro 4.16:** Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase O-G por el método de validación cruzada estratificada de 20 pliegues.

Modelo	AUC	CA	F1	Precisión	Recall
kNN	0.945	0.887	0.830	0.800	0.863
SVM	0.983	0.931	0.893	0.885	0.902
Red neuronal	0.988	0.937	0.904	0.887	0.922
Regresión logística	0.993	0.950	0.923	0.906	0.941

Al resumir las tres clases en el promedio AOC (cuadro 4.17) los resultados más destacables se dan en la métrica AUC, de esta manera, la red SqueezeNet con 1000 descriptores de imagen tienen como mejor desempeño la separabilidad de clases. Los resultados en las demás métricas son bastante similares cuando se trata del mismo modelo predictivo. Se observa una disminución en los puntajes generales comparados con la red Inception V3, solo que ahora el modelo con mejores resultados fue el de red neuronal. Así, en la implementación de la CNN SqueezeNet, el mejor modelo para la clasificación de acoplamientos de sistemas fisiológicos es el de red neuronal al ser entrenado por la validación cruzada estratificada.

**Cuadro 4.17:** Tabla de resultados con el promedio de los resultados AOC en cada clase por el método de validación cruzada estratificada de 20 pliegues.

Modelo	AUC	CA	F1	Precisión	Recall
kNN	0.894	0.767	0.765	0.764	0.767
SVM	0.937	0.811	0.811	0.810	0.811
Red neuronal	0.963	0.843	0.842	0.843	0.843
Regresión logística	0.955	0.836	0.836	0.840	0.836

Nuevamente a partir de la matriz Bayessiana (cuadro 4.18), podemos notar que la probabilidad de que la regresión logística obtenga mejores resultados que los modelos kNN y SVM, es alta. En el caso de la red neuronal, una probabilidad de 0.547 por encima de la regresión logística, no es determinante al afirmar que este modelo obtendrá mejores resultados. Así, considerando los resultados previos, podemos determinar que en general, la regresión logística es el modelo de aprendizaje supervisado con mejor desempeño en la metodología implementada cuando los modelos son entrenados por el método de validación cruzada estratificada.

**Cuadro 4.18:** Matriz Bayessiana con el análisis entre modelos para el método de validación cruzada estratificada de 20 pliegues de la red SqueezeNet.

	kNN	SVM	Red neuronal	Regresión logística
kNN		0.015	0.007	0.005
SVM	0.985		0.166	0.153
Red neuronal	0.993	0.834		0.453
Regresión logística	0.995	0.847	0.547	

### 4.2.2. Leave One Out

Las matrices de confusión correspondientes a cada modelo predictivo entrenados por el método de leave one out, se presentan en el cuadro 4.19. Nuevamente, podemos notar que el número de errores en todos los modelos aumentan al utilizar esta red.

**Cuadro 4.19:** Matrices de confusión para cada modelo predictivo por el método de validación cruzada de 20 pliegues de la red SqueezeNet.

Regresión logística:

		Predicha			$\Sigma$
		G-B	O-B	O-G	
Real	G-B	41	11	1	53
	O-B	8	44	3	55
	O-G	0	3	48	51
$\Sigma$		59	48	52	159

Red neuronal:

		Predicha			$\Sigma$
		G-B	O-B	O-G	
Real	G-B	46	6	1	53
	O-B	7	43	5	55
	O-G	1	3	47	51
	$\Sigma$	59	48	52	159

kNN:

		Predicha			$\Sigma$
		G-B	O-B	O-G	
Real	G-B	44	7	2	53
	O-B	10	36	9	55
	O-G	1	4	46	51
	$\Sigma$	59	48	52	159

SVM:

		Predicha			$\Sigma$
		G-B	O-B	O-G	
Real	G-B	44	9	0	53
	O-B	9	40	6	55
	O-G	2	3	46	51
	$\Sigma$	59	48	52	159

Por el método de leave one out (validación cruzada con 159 pliegues) se obtiene para cada modelo los resultados siguientes en las métricas.

Al comparar los resultados con la red Inception de la clase G-B (cuadros 4.20 y 4.9), es notoria una disminución en los puntajes de casi todas las métricas excepto en la precisión de los modelos kNN, SVM y red neuronal, dicha disminución no es tan drástica como en el método de entrenamiento pasado (lo cual era de esperarse debido a que estamos entrenando con un conjunto más grande de muestra a cada modelo, que cuando se hace con la validación cruzada).

**Cuadro 4.20:** Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase G-B por el método de leave one out.

Modelo	CA	F1	Precisión	Recall
kNN	0.874	0.815	0.800	0.830
SVM	0.874	0.815	0.800	0.30
Red neuronal	0.906	0.860	0.852	0.868
Regresión logística	0.874	0.804	0.837	0.774

La disminución en los puntajes de las métricas se hace más notoria en la clase O-B (cuadro 4.21). A diferencia del acoplamiento G-B, este no obtuvo mejores puntajes en la precisión, de hecho, fue la métrica que más redujo su puntaje si lo comparamos con el cuadro 4.18. Es esta misma razón por la que el puntaje F1 también disminuye.

**Cuadro 4.21:** Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase O-B por el método de leave one out.

Modelo	CA	F1	Precisión	Recall
kNN	0.811	0.706	0.766	0.655
SVM	0.830	0.748	0.769	0.727
Red neuronal	0.868	0.804	0.827	0.782
Regresión logística	0.843	0.779	0.759	0.800

Para la última etiqueta de clase, O-G (cuadro 4.22), los resultados comparados con el cuadro 4.11, nuevamente presenta un disminución, pero no tan grande como en las otras clases. Este acoplamiento se caracteriza por su alto puntaje en recall, esto significa que los modelos tienen un buen desempeño en la clasificación de positivos para este acoplamiento.

**Cuadro 4.22:** Tabla de resultados de la clasificación de las gráficas de recurrencia de la clase O-G por el método de leave one out.

Modelo	CA	F1	Precisión	Recall
kNN	0.899	0.852	0.807	0.902
SVM	0.931	0.893	0.885	0.902
Red neuronal	0.937	0.904	0.887	0.922
Regresión logística	0.956	0.932	0.923	0.941

Los resultados AOC (cuadro 4.23), al igual que en la validación cruzada (cuadro 4.18), la red neuronal obtuvo los mejores puntajes en todas las métricas de todos los modelos. En la implementación de la CNN SqueezeNet, el modelo de aprendizaje automático supervisado con mejor desempeño es el de red neuronal.

**Cuadro 4.23:** Tabla de resultados con el promedio de los resultados AOC en cada clase por el método leave one out.

Modelo	AUC	CA	F1	Precisión	Recall
kNN	0.895	0.792	0.789	0.790	0.792
SVM	0.936	0.818	0.817	0.816	0.818
Red neuronal	0.965	0.855	0.855	0.854	0.855
Regresión logística	0.955	0.836	0.836	0.837	0.836

Nuevamente, obteniendo los mayores puntajes en las 5 métricas, el modelo de red neuronal.

Los resultados en promedio son adecuados considerando el número de datos disponibles. La CNN Inception V3 de Google permite obtener mejores métricas de evaluación que la CNN SqueezeNet; esto debido al número de vectores con los que se representa cada diagrama. Como las gráficas son arreglos matriciales de  $15000 \times 15000$  entradas, al reducir los arreglos a vectores de 1000 entradas existe una importante pérdida de información lo que ocasiona que la clasificación sea más complicada para los modelos predictivos.

En la implementación de un algoritmo para la predicción de estados de acoplamiento, se busca un puntaje alto en AUC y en recall. Un puntaje alto en AUC representa una muy buena separabilidad y de esta manera una muy buena diferenciación en los estados acoplados y no acoplados. A la vez, como la mayoría de diagnósticos clínicos, se busca el menor número de falsos negativos, ya que cada diagnóstico está asociado a una patología, un falso negativo permitirá la evolución de dicha patología y podría resultar mortal. La curva ROC está construida a través de la FPR (Tasa de falsos negativos) por lo que al obtener un puntaje alto en AUC, implica una FPR baja. El recall no solo identifica todas las muestras positivas, sino que limita el número de falsos negativos, por lo que un recall alto es necesario en un modelo predictivo de este tipo.

## 5 Conclusiones

Sabemos que todo proceso fisiológico esta compuesto por una intrincada red de reacciones bioquímicas, por lo que comprender la compleja dinámica que subyace a estos procesos es de vital importancia, ya que desajustes en estos procesos están comúnmente relacionados con la emergencia de enfermedades complejas. Por lo anterior, el uso de herramientas computacionales y matemáticas es necesario para el análisis de estos complejos mecanismos bioquímicos.

La gráfica de recurrencia es una herramienta matemática que permite el estudio de estos mecanismos. Esta herramienta permite exhibir patrones en las series temporales, proporcionando información valiosa para su análisis, por lo que estas gráficas se pueden utilizar para entrenar modelos de aprendizaje automático supervisado para clasificar el tipo de osciladores acoplados.

La extracción de descriptores a partir de las gráficas de recurrencia de osciladores acoplados empleando CNNs previamente entrenadas, permite entrenar eficientemente modelos de aprendizaje supervisado de clasificación y obtener valores de las métricas de evaluación con alta precisión . En particular, la implementación de las CNNs Inception V3 y SqueezeNet para la extracción de descriptores a partir de las gráficas de recurrencia de osciladores bioquímicos acoplados permite entrenar, validar y probar modelos de clasificación eficientemente y a su vez obtener resultados sobresalientes; en especial en el modelo de regresión logística implementando la red Inception V3, obteniendo como resultados promedio en las métricas de evaluación 92.62 % y 93.02 % de exactitud al ser entrenado por el método de validación cruzada estratificada y leave one out respectivamente.

La metodología propuesta en este trabajo puede extenderse al campo experimental para facilitar la determinación de estados de sincronización y desincronización en complejos mecanismos bioquímicos y fisiológicos reales, para reconocer una posible correlación entre estos estados y la emergencia de diferentes enfermedades complejas.



## **Agradecimientos**

En primer lugar quiero agradecer a mi asesor, el M.C. Jesús Andrés Arzola Flores, quien con sus conocimientos y apoyo me guió a través de cada una de las etapas de este proyecto y que su guía fue de vital importancia en la realización de esta tesis.

También quiero agradecer a la Benemérita Universidad Autónoma de Puebla por brindarme todos los recursos, herramientas y acompañamiento para poder desempeñarme como físico aplicado.

Por último, quiero agradecer a mi familia y amigos por todo el apoyo incondicional, por la motivación para cumplir mis metas, por todo el amor recibido y por ser una parte importante en mi formación. En especial, quiero agradecer a mis padres por ser los mejores ejemplos de vida. A mi padre por enseñarme que con trabajo duro y disciplina, los sueños sí se cumplen; a mi madre por compartirme su empatía con los demás y su compromiso con la gente que ama. Son estas dos cualidades aunadas al hecho de que siempre hayan creído en mí, los motivos por los cuales sigo luchando todos los días.



# Bibliografía

- [1] G. Adomian. “The diffusion-Brusselator equation”. En: *Computers Mathematics with Applications* 29.5 (1995), págs. 1-3. ISSN: 0898-1221. DOI: [https://doi.org/10.1016/0898-1221\(94\)00244-F](https://doi.org/10.1016/0898-1221(94)00244-F). URL: <https://www.sciencedirect.com/science/article/pii/089812219400244F>.
- [2] Uri M. Ascher y Linda R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. 1st. USA: Society for Industrial y Applied Mathematics, 1998. ISBN: 0898714125.
- [3] Raheel Chaudhry y Matthew Varacallo. *Biochemistry, Glycolysis*. eng. Treasure Island (FL), 2021.
- [4] William W. Chen, Mario Niepel y Peter K. Sorger. “Classic and contemporary approaches to modeling biochemical reactions”. En: *Genes Development* 24.17 (2010), págs. 1861-1875. DOI: 10.1101/gad.1945410. eprint: <http://genesdev.cshlp.org/content/24/17/1861.full.pdf+html>. URL: <http://genesdev.cshlp.org/content/24/17/1861.abstract>.
- [5] Giorgio Corani y Alessio Benavoli. “A Bayesian approach for comparing cross-validated algorithms on multiple data sets”. En: *Machine Learning* 100.2 (2015), págs. 285-304. ISSN: 1573-0565. DOI: 10.1007/s10994-015-5486-z. URL: <https://doi.org/10.1007/s10994-015-5486-z>.
- [6] Luciano da F. Costa, Francisco A. Rodrigues y Alexandre S. Cristino. “Complex networks: the key to systems biology”. en. En: *Genetics and Molecular Biology* 31 (2008), págs. 591-601. ISSN: 1415-4757. URL: [http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1415-47572008000400001&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1415-47572008000400001&nrm=iso).
- [7] Guillermo Davila-Rascon, José Vallejo y Antonio Morante. *Synchronization of dynamical systems: an approach using a Computer Algebra System*. Sep. de 2018. DOI: 10.13140/RG.2.2.12662.45127.
- [8] J. Demšar y Blaz Zupan. “Orange: Data mining fruitful and fun - A historical perspective”. En: *Informatica (Slovenia)* 37 (mar. de 2013), págs. 55-60.
- [9] Janez Demšar y col. “Orange: Data Mining Toolbox in Python”. En: *J. Mach. Learn. Res.* 14.1 (ene. de 2013), 2349â2353. ISSN: 1532-4435.
- [10] Google Developers. *Machine Learning Crash Course*. sitio web. URL: <https://developers.google.com/machine-learning/crash-course>.

- [11] J.-P. Eckmann, S. Oliffson Kamphorst y D. Ruelle. “Recurrence Plots of Dynamical Systems”. En: *Europhysics Letters (EPL)* 4.9 (1987), págs. 973-977. DOI: 10.1209/0295-5075/4/9/004. URL: <https://doi.org/10.1209/0295-5075/4/9/004>.
- [12] D. Edelson, R. J. Field y R. M. Noyes. “Mechanistic details of the Belousov â Zhabotinskii oscillations”. En: *International Journal of Chemical Kinetics* 7.3 (1975), págs. 417-432. DOI: <https://doi.org/10.1002/kin.550070309>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/kin.550070309>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/kin.550070309>.
- [13] Irving R. Epstein, John A. Pojman y Oliver Steinbock. “Introduction: Self-organization in nonequilibrium chemical systems”. En: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 16.3 (2006), pág. 037101. DOI: 10.1063/1.2354477. eprint: <https://aip.scitation.org/doi/pdf/10.1063/1.2354477>. URL: <https://aip.scitation.org/doi/abs/10.1063/1.2354477>.
- [14] Leon Glass. “Synchronization and rhythmic processes in physiology”. En: *Nature* 410.6825 (2001), págs. 277-284. ISSN: 1476-4687. DOI: 10.1038/35065745. URL: <https://doi.org/10.1038/35065745>.
- [15] Stanley I. Grossman. “11 - Ordinary Differential Equations”. En: *Multivariable Calculus, Linear Algebra, and Differential Equations (Second Edition)*. Ed. por STANLEY I. GROSSMAN. Second Edition. Academic Press, 1986, págs. 619-718. ISBN: 978-0-12-304380-1. DOI: <https://doi.org/10.1016/B978-0-12-304380-1.50015-4>. URL: <https://www.sciencedirect.com/science/article/pii/B9780123043801500154>.
- [16] Kornber H. “Krebs and his trinity of cycles”. En: *Nat Rev Mol Cell Biol* 1.3 (December 2000), págs. 225-233.
- [17] J. F. Heagy, T. L. Carroll y L. M. Pecora. “Desynchronization by periodic orbits”. En: *Phys. Rev. E* 52 (2 1995), R1253-R1256. DOI: 10.1103/PhysRevE.52.R1253. URL: <https://link.aps.org/doi/10.1103/PhysRevE.52.R1253>.
- [18] Chih-wei Hsu, Chih-chung Chang y Chih-Jen Lin. “A Practical Guide to Support Vector Classification Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin”. En: (nov. de 2003).
- [19] J. L. Hudson y J. C. Mankin. “Chaos in the Belousov â Zhabotinskii reaction”. En: *The Journal of Chemical Physics* 74.11 (1981), págs. 6171-6177. DOI: 10.1063/1.441007. eprint: <https://doi.org/10.1063/1.441007>. URL: <https://doi.org/10.1063/1.441007>.
- [20] Forrest Iandola y col. “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size”. En: (feb. de 2016).
- [21] *IDEA - Internet Differential Equations Activities*. Washington State University. URL: <http://www.idea.wsu.edu/OscilChem/#BrusselatorModel>.

- [22] David James. *Introduction to Machine Learning with Python: A Guide for Beginners in Data Science*. 1st. North Charleston, SC, USA: CreateSpace Independent Publishing Platform, 2018. ISBN: 1726230872.
- [23] James Keener y James Sneyd. *Mathematical Physiology*. Berlin, Heidelberg: Springer-Verlag, 1998. ISBN: 0387983813.
- [24] Istvan Kiss y col. "Travelling waves in the Oregonator model for the BZ reaction". En: *IMA Journal of Applied Mathematics - IMA J APPL MATH* 74 (ago. de 2009), págs. 622-643. DOI: 10.1093/imamat/hxp009.
- [25] J. Laaksonen y E. Oja. "Classification with learning k-nearest neighbors". En: *Proceedings of International Conference on Neural Networks (ICNN'96)*. Vol. 3. 1996, 1480-1483 vol.3. DOI: 10.1109/ICNN.1996.549118.
- [26] Jun Ma y col. "Complete synchronization, phase synchronization and parameters estimation in a realistic chaotic system". En: *Communications in Nonlinear Science and Numerical Simulation* 16.9 (2011), págs. 3770-3785. ISSN: 1007-5704. DOI: <https://doi.org/10.1016/j.cnsns.2010.12.030>. URL: <https://www.sciencedirect.com/science/article/pii/S1007570410006490>.
- [27] Jesús Martínez y col. "Sincronización maestro-esclavo en Sistemas Dinámicos Lineales y No Lineales". En: 49 (dic. de 2016), págs. 132-140.
- [28] Thomas M. Mitchell. *Machine Learning*. 1.<sup>a</sup> ed. USA: McGraw-Hill, Inc., 1997. ISBN: 0070428077.
- [29] Dane Morgan y Ryan Jacobs. "Opportunities and Challenges for Machine Learning in Materials Science". En: *Annual Review of Materials Research* 50.1 (2020), págs. 71-103. DOI: 10.1146/annurev-matsci-070218-010015. eprint: <https://doi.org/10.1146/annurev-matsci-070218-010015>. URL: <https://doi.org/10.1146/annurev-matsci-070218-010015>.
- [30] Andreas C. Muller y Sarah Guido. *Introduction to machine learning with Python : a guide for data scientists*. English. 2017. URL: <http://www.dawsonera.com/depp/reader/protected/external/AbstractView/S9781449369903>.
- [31] Francesco Nardelli y col. En.
- [32] Sorge Oporto-Almaraz y Gonzalo Marcelo Ramírez vila. "Análisis del comportamiento oscilatorio de las concentraciones en la reacción de Belousov - Zhabotinsky". En: *Revista Boliviana de Química* 30 (ene. de 2013), págs. 102-114.
- [33] Raúl Rojas. *Neural Networks: A Systematic Introduction*. Berlin, Heidelberg: Springer-Verlag, 1996. ISBN: 3540605053.
- [34] David E. Rumelhart, Geoffrey E. Hinton y Ronald J. Williams. "Learning representations by back-propagating errors". En: *Nature* 323.6088 (1986), págs. 533-536. ISSN: 1476-4687. DOI: 10.1038/323533a0. URL: <https://doi.org/10.1038/323533a0>.

- [35] P Rustin y col. “Inborn errors of the Krebs cycle: a group of unusual mitochondrial diseases in human”. En: *Biochim Biophys Acta* 1361.2 (1997 Aug 22), págs. 185-97.
- [36] R. Obaya J. Rojo S. Novo. *Ecuaciones y sistemas diferenciales*. MacGraw Hill, 1995.
- [37] Francesc Sagués e Irving R. Epstein. “Nonlinear chemical dynamics”. En: *Dalton Trans.* (7 2003), págs. 1201-1217. DOI: 10.1039/B210932H. URL: <http://dx.doi.org/10.1039/B210932H>.
- [38] “Logistic Regression”. En: *Encyclopedia of Machine Learning*. Ed. por Claude Sammut y Geoffrey I. Webb. Boston, MA: Springer US, 2010, págs. 631-631. ISBN: 978-0-387-30164-8. DOI: 10.1007/978-0-387-30164-8\_493. URL: [https://doi.org/10.1007/978-0-387-30164-8\\_493](https://doi.org/10.1007/978-0-387-30164-8_493).
- [39] K. A. Schafer. “The Cell Cycle: A Review”. En: *Veterinary Pathology* 35.6 (1998). PMID: 9823588, págs. 461-478. DOI: 10.1177/030098589803500601. eprint: <https://doi.org/10.1177/030098589803500601>. URL: <https://doi.org/10.1177/030098589803500601>.
- [40] A. / Vishwanathan S. Smola. *Introduction to Machine Learning*. Cambridge University Press, 2008.
- [41] Krishnamachari Sriram y Samuel Bernard. “Complex dynamics in the Oregonator model with linear delayed feedback”. En: *Chaos (Woodbury, N.Y.)* 18 (jul. de 2008), pág. 023126. DOI: 10.1063/1.2937015.
- [42] Steven Strogatz. *Nonlinear Dynamics and Chaos: with Applications to Physics, Biology, Chemistry, and Engineering*. Westview Press, 2015.
- [43] Christian Szegedy y col. “Rethinking the Inception Architecture for Computer Vision”. En: jun. de 2016. DOI: 10.1109/CVPR.2016.308.
- [44] Arunas Tamasevicius, A. Namajunas y Antanas Cenys. “Simple 4D chaotic oscillator”. En: *Electronics Letters* 32 (jun. de 1996), págs. 957 -958. DOI: 10.1049/e1:19960630.
- [45] G. Vidal. “Sincronización y control de sistemas dinámicos en régimen de caos espacio-temporal.” Tesis doct. Universidad de Navarra España, 2010.
- [46] Niamh Williams y Luke O’Neill. “A Role for the Krebs Cycle Intermediate Citrate in Metabolic Reprogramming in Innate Immunity and Inflammation”. En: *Frontiers in Immunology* 9 (feb. de 2018). DOI: 10.3389/fimmu.2018.00141.
- [47] Abdallah Yousif, Zhendong Niu y Ally S. Nyamawe. “Citation Classification Using Multitask Convolutional Neural Network Model”. En: *Knowledge Science, Engineering and Management*. Ed. por Weiru Liu, Fausto Giunchiglia y Bo Yang. Cham: Springer International Publishing, 2018, págs. 232-243. ISBN: 978-3-319-99247-1.

- [48] Anatol M. Zhabotinsky. “A history of chemical oscillations and waves”. En: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 1.4 (1991), págs. 379-386. DOI: 10.1063/1.165848. eprint: <https://doi.org/10.1063/1.165848>. URL: <https://doi.org/10.1063/1.165848>.



# Nomenclatura

$\Phi$	Función de incremento
$\epsilon$	Umbral de las gráficas de recurrencia
$\sigma_1$	Concentración de ATP
$\sigma_2$	Concentración de ADP
AUC	Área de bajo de la curva ROC
CA	Exactitud
FN	Falso negativo
FP	Falso positivo
FPR	Tasa de falsos positivos
G-B	Glucólisis y oscilaciones glucolíticas con brusselator
ML	Aprendizaje automático
O-B	Oregonator con brusselator
O-G	Glucólisis y oscilaciones glucolíticas con oregonator
ROC	Curva característica operativa del receptor
SVM	Máquina de vectores de soporte
TN	Verdadero negativo
TP	Verdadero positivo
TPR	Tasa de verdaderos positivos
$k$	Parámetro de sincronización
kNN	k-Nearest Neighbors
$x$	Concentración adimensional de $\text{HBrO}_2$

- y Especie intermedia 2 en brusselator
- z Especie intermedia 1 en brusselator
- z Forma oxidada del catalizador adimensional en oregonator