



UNIVERSIDAD DE CHILE

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

DEPARTAMENTO DE INGENIERÍA QUÍMICA, BIOTECONOLOGÍA
Y MATERIALES

**ESTUDIO Y APLICACIÓN DE ALGORITMOS DE
MACHINE LEARNING PARA EL DISEÑO DE UN
SIMULADOR PREDICTIVO DEL COMPORTAMIENTO
DINÁMICO DE ESPESADORES DE RELAVES**

**MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL
QUÍMICO**

IGNACIO ALEXANDER VALENZUELA GONZÁLEZ

**PROFESOR GUÍA:
JOSÉ SALGADO HERRERA**

**MIEMBROS DE LA COMISIÓN:
FRANCISCO GRACIA CAROCA
ÁLVARO OLIVERA-NAPPA
CHRISTIAN IHLE BASCUÑÁN**

**SANTIAGO DE CHILE
2023**

ESTUDIO Y APLICACIÓN DE ALGORITMOS DE MACHINE LEARNING PARA EL DISEÑO DE UN SIMULADOR PREDICTIVO DEL COMPORTAMIENTO DINÁMICO DE ESPESADORES DE RELAVES

Una de las problemáticas más importante en los procesos mineros es aprovechar al máximo el uso de agua, y una de las operaciones unitarias con mayor importancia en este ámbito es el espesamiento de relaves. A. Los relaves entran con aproximadamente un 20% a 30% de concentración de sólidos, y sus salidas corresponde a agua recuperada, y relaves al 50% de sólidos, por lo que se hace necesario desarrollar nuevas tecnologías que permitan optimizar las operaciones involucradas en estos procesos.

El objetivo de este trabajo es estudiar el desempeño de algoritmos del área de Inteligencia Artificial, aplicados a un sistema de espesamiento de relaves, en la obtención de los parámetros de las ecuaciones utilizadas para la modelación fenomenológica de dicho proceso. Para esto se disponen de datos experimentales de los parámetros relacionados a las funciones constitutivas que describen el proceso de sedimentación de distintas muestras de relaves dentro de un espesador. Se diseñó modelos basados en 3 tipos de algoritmos de Machine Learning: un algoritmo de programación genética y 2 tipos de redes neuronales artificiales (recurrentes y convolucionales). Estos modelos se implementaron para cada uno de los parámetros respectivos, teniendo como variables de entrada los porcentajes de las principales arcillas presentes en las muestras, junto al porcentaje de material fino en los relaves.

Sobre la programación genética, se obtuvo 8 expresiones matemáticas que describen los parámetros de las funciones constitutivas asociadas a la teoría de sedimentación. El coeficiente de correlación promedio de los modelos basados en este algoritmo es de $R^2 = 0,7951$ y el error cuadrático medio promedio de $RMSE = 0.1414$. El tiempo estimado de entrenamiento de cada modelo fue de 3 horas y 25 minutos y el de ejecución es de 3,5 segundos. Por otro lado, se obtuvo 8 los modelos basados en cada tipo de red neuronal a estudiar. Para el caso de las redes recurrentes se obtuvo un coeficiente de correlación promedio de $R^2 = 0,9702$, un error cuadrático medio $RMSE = 0,0457$, tiempo de entrenamiento y ejecución de aproximadamente 5 horas y 4,0 segundos, respectivamente. En cuanto a las redes convolucionales, para el coeficiente de correlación se tiene $R^2 = 0,9613$, error cuadrático medio $RMSE = 0,0494$, tiempo de entrenamiento de 5 horas y de ejecución de 4,1 segundos.

Del análisis realizado, se concluye que a pesar de que los algoritmos basados en programación genética utilizan menos recursos en comparación a las redes neuronales, éstas últimas poseen una mayor capacidad de predicción, destacando a las redes recurrentes en la simulación global de los datos, y a las redes neuronales convolucionales para la detección y mejor predicción de las muestras con bajo nivel de arcillas y material fino. Por último, en el trabajo se planteó una metodología para desarrollar modelos empíricos que puede utilizarse en otros sistemas similares.

A mis padres,
Gracias por todo.

Agradecimientos

En este fin de ciclo quiero tomarme un tiempo para agradecer a todas las personas que formaron parte de mi camino por la vida universitaria. Sin embargo, en primer lugar, debo agradecer a mis profesores de enseñanza media, que fueron los primeros en guiarme en mis pasiones, en mis habilidades y complementaron inmensamente mi educación valórica. A mi profesora de física y matemáticas Wilma Fuentes, que supo canalizar todas las inquietudes científicas y académicas de aquel distante estudiante, y a mi profesor de lenguaje Jorge Román, quien me enseñó una mirada integral de la vida y a no dejar de lado las demás ramas del conocimiento.

Agradecer a todas las hermosas personas que conocí durante estos años en la universidad. Mis amigos de Plan Común, con quienes tengo muchos recuerdos inolvidables dando nuestros primeros pasos en la facultad. A mis compañeros de generación y generaciones cercanas del departamento, que me enseñaron muchas cosas y siempre tuve grandes conversaciones memorables.

También quiero agradecer a las personas directamente involucradas en este trabajo, principalmente a Nicolás Fuenzalida, quien confió en mi para la elaboración original de este proyecto, y me apoya constantemente. Además, quiero agradecer al profesor Cristian Salgado, quien pudo ayudarme de muchas formas en el desarrollo de esta memoria, a los profesores de la comisión que me guiaron y ayudaron en las consultas pertinentes, y finalmente a todos los docentes que me enseñaron durante la carrera, especialmente a los de IQBT, por sus enseñanzas y guía profesional.

Finalmente, dedicarle las últimas líneas a mi familia, a mi padre y madre que les debo todo lo que soy, por su apoyo, su amor y preocupación en todos los ámbitos de mi vida. A mi hermano, por sus conversaciones profundas y decisivas. Sin ustedes yo no sería el mismo y tampoco hubiera logrado este viaje.

Tabla de Contenido

1. Introducción	1
1.1 Antecedentes Generales	1
1.2 Motivación	2
1.3 Descripción del proyecto	3
1.4 Justificación	4
2. Objetivos y limitaciones	6
2.1 Objetivos	6
2.2 Limitaciones	6
3. Marco teórico	8
3.1 Proceso del concentrado de cobre y molibdeno	8
3.2 Caracterización del espesador de relaves estudiado	10
3.3 Modelo de espesador de relaves	12
3.4 Definición de funciones constitutivas	15
3.5 Efecto de arcillas en las funciones constitutivas	16
3.6 Inteligencia Artificial en procesos mineros	19
3.7 Análisis exploratorio de datos	19
3.8 Algoritmos de Machine Learning	22
3.9 Validación de resultados obtenidos de algoritmos de Machine Learning	31
4. Metodología	34
4.1 Definición de variables de entrada y salida de algoritmos de Machine Learning	35
4.2 Entrenamiento de redes neuronales	37
4.3 Implementación del algoritmo de programación genética	39
4.4 Validación de resultados	40
5. Resultados y discusión	42
5.1 Dispersión y Variabilidad de inputs y targets	42
5.2 Implementación de algoritmo de programación genética	45

5.3	Implementación de redes neuronales.....	54
5.4	Resultados obtenidos por los modelos de Machine Learning	70
5.5	Comparación modelos: criterios estadísticos	81
5.6	Comparación de modelos	83
5.7	Comentarios Generales.....	112
6.	Conclusiones.....	115
	Bibliografía	119
	Anexos	125
	Anexo A: Funciones de activación para el diseño de redes neuronales artificiales ...	125
	Anexo B: Resultados detallados algoritmo de programación genética	126
	Anexo C: Funciones constitutivas para cada muestra de relave	131

Índice de Tablas

Tabla 1: Funciones constitutivas de estudio.	3
Tabla 2: Caracterización del espesor de relaves estudiado.	12
Tabla 3: Supuestos utilizados para el diseño del modelo.....	12
Tabla 4: Valores de variables de entrada de las muestras utilizadas.....	36
Tabla 5: Parámetros utilizados en el algoritmo de red neuronal.	37
Tabla 6: Parámetros utilizados en el algoritmo de programación genética	40
Tabla 7: Valores normalizados obtenidos por cada modelo. Pre-exponencial función batch de Kynch.....	71
Tabla 8: Valores normalizados obtenidos por cada modelo. Exponente función batch de Kynch.	71
Tabla 9: Valores normalizados obtenidos por cada modelo. Pre-exponencial función estrés efectivo del sólido.	73
Tabla 10: Valores normalizados obtenidos por cada modelo. Exponente función estrés efectivo del sólido.	74
Tabla 11: Valores normalizados obtenidos por cada modelo. Coeficiente cuadrático función de dosificación.	76
Tabla 12: Valores normalizados obtenidos por cada modelo. Coeficiente lineal función de dosificación.	77
Tabla 13: Valores normalizados obtenidos por cada modelo. Coeficiente libre función de dosificación.	77
Tabla 14: Valores normalizados obtenidos por cada modelo. Concentración crítica.	80
Tabla 15: Coeficientes estadísticos obtenidos para cada modelo diseñado. Función batch de Kynch.	81
Tabla 16: Coeficientes estadísticos obtenidos para cada modelo diseñado. Función estrés efectivo del sólido.	82
Tabla 17: Coeficientes estadísticos obtenidos para cada modelo diseñado. Función de dosificación.	82
Tabla 18: Coeficientes estadísticos obtenidos para cada modelo diseñado. Concentración crítica.....	83
Tabla 19: Tiempos de cómputo de los modelos diseñados.	112
Tabla 20: Funciones comunes de activación de neuronas.	125

Índice de Ilustraciones

Figura 1: Representación gráfica del equipo simulado.....	3
Figura 2: Proceso del cobre.....	8
Figura 3: Diagrama de la geometría del espesador de estudio.....	11
Figura 4: Perfil de concentración de sólidos para la muestra R01.....	14
Figura 5: Perfil de concentración de sólidos para la muestra R03.....	14
Figura 6: Perfil de concentración de sólidos para la muestra R04.....	15
Figura 7: Efecto de la dosificación de floculante en la recuperación y velocidad de sedimentación.....	18
Figura 8: Representación de un diagrama de cajas y bigotes.....	21
Figura 9: Diagrama de flujo del algoritmo de programación genética.....	23
Figura 10: Diagrama de una red neuronal artificial.....	25
Figura 11: Diagrama de red neuronal recurrente.....	26
Figura 12: Diagrama de red neuronal convolucional.....	28
Figura 13: Etapas de reducción y convolución detalladas de una red neuronal convolucional.....	28
Figura 14: Diagrama de metodología utilizada.....	34
Figura 15: Diagrama de flujo de variables en algoritmos de Machine Learning.....	37
Figura 16: Diagrama de la red neuronal recurrente diseñada.....	39
Figura 17: Diagrama de red neuronal convolucional diseñada.....	39
Figura 18: Dispersión de las variables de entrada para los algoritmos de Machine Learning.....	42
Figura 19: Dispersión del target para los algoritmos de Machine Learning.....	44
Figura 20: Variables involucradas en la modelación de A a partir de GP.....	46
Figura 21: Variables involucradas en la modelación de b a partir de GP.....	47
Figura 22: Variables involucradas en la modelación de C a partir de GP.....	48
Figura 23: Variables involucradas en la modelación de d a partir de GP.....	49
Figura 24: Variables involucradas en la modelación de p_1 a partir de GP.....	50
Figura 25: Variables involucradas en la modelación de p_2 a partir de GP.....	51
Figura 26: Variables involucradas en la modelación de p_3 a partir de GP.....	52
Figura 27: Variables involucradas en la modelación de ϕ_c a partir de GP.....	53
Figura 28: Variables involucradas en la modelación de A a partir de RNN.....	54
Figura 29: Variables involucradas en la modelación de b a partir de RNN.....	55
Figura 30: Variables involucradas en la modelación de C a partir de RNN.....	56
Figura 31: Variables involucradas en la modelación de d a partir de RNN.....	57

Figura 32: Variables involucradas en la modelación de p_1 a partir de RNN.	58
Figura 33: Variables involucradas en la modelación de p_2 a partir de RNN.	59
Figura 34: Variables involucradas en la modelación de p_3 a partir de RNN.	60
Figura 35: Variables involucradas en la modelación de ϕC a partir de RNN.	61
Figura 36: Variables involucradas en la modelación de A a partir de CNN.	62
Figura 37: Variables involucradas en la modelación de b a partir de CNN.	63
Figura 38: Variables involucradas en la modelación de C a partir de CNN.	64
Figura 39: Variables involucradas en la modelación de d a partir de CNN.	65
Figura 40: Variables involucradas en la modelación de p_1 a partir de CNN.	66
Figura 41: Variables involucradas en la modelación de p_2 a partir de CNN.	67
Figura 42: Variables involucradas en la modelación de p_3 a partir de CNN.	68
Figura 43: Variables involucradas en la modelación de ϕc a partir de CNN.	69
Figura 44: Valores experimentales y predichos para A.	72
Figura 45: Valores experimentales y predichos para b.	73
Figura 46: Valores experimentales y predichos para C.	75
Figura 47: Valores experimentales y predichos para d.	75
Figura 48: Valores experimentales y predichos para p_1	78
Figura 49: Valores experimentales y predichos para p_2	79
Figura 50: Valores experimentales y predichos para p_3	79
Figura 51: Valores experimentales y predichos para ϕc	81
Figura 52: Simulación del coeficiente A en función del porcentaje de caolinita.	83
Figura 53: Simulación del coeficiente b en función del porcentaje de caolinita.	84
Figura 54: Simulación del coeficiente A en función del porcentaje de clorita.	85
Figura 55: Simulación del coeficiente b en función del porcentaje de clorita.	86
Figura 56: Simulación del coeficiente A en función del porcentaje de esmectita.	87
Figura 57: Simulación del coeficiente b en función del porcentaje de esmectita.	87
Figura 58: Simulación del coeficiente A en función del porcentaje de moscovita.	89
Figura 59: Simulación del coeficiente b en función del porcentaje de esmectita.	89
Figura 60: Simulación del coeficiente A en función de D_2	90
Figura 61: Simulación del coeficiente b en función de D_2	90
Figura 62: Simulación del coeficiente C en función del porcentaje de caolinita.	91
Figura 63: Simulación del coeficiente d en función del porcentaje de caolinita.	92
Figura 64: Simulación del coeficiente C en función del porcentaje de clorita.	93
Figura 65: Simulación del coeficiente d en función del porcentaje de clorita.	93
Figura 66: Simulación del coeficiente C en función del porcentaje de esmectita.	94
Figura 67: Simulación del coeficiente d en función del porcentaje de esmectita.	94
Figura 68: Simulación del coeficiente C en función del porcentaje de moscovita.	95
Figura 69: Simulación del coeficiente d en función del porcentaje de moscovita.	96

Figura 70: Simulación del coeficiente C en función de $D2$.	97
Figura 71: Simulación del coeficiente d en función de $D2$.	97
Figura 72: Simulación del coeficiente $p1$ en función del porcentaje de caolinita.	98
Figura 73: Simulación del coeficiente $p2$ en función del porcentaje de caolinita.	98
Figura 74: Simulación del coeficiente $p3$ en función del porcentaje de caolinita.	99
Figura 75: Simulación del coeficiente $p1$ en función del porcentaje de clorita.	100
Figura 76: Simulación del coeficiente $p2$ en función del porcentaje de clorita.	100
Figura 77: Simulación del coeficiente $p3$ en función del porcentaje de clorita.	101
Figura 78: Simulación del coeficiente $p1$ en función del porcentaje de esmectita.	102
Figura 79: Simulación del coeficiente $p2$ en función del porcentaje de esmectita.	102
Figura 80: Simulación del coeficiente $p3$ en función del porcentaje de esmectita.	103
Figura 81: Simulación del coeficiente $p1$ en función del porcentaje de moscovita.	104
Figura 82: Simulación del coeficiente $p2$ en función del porcentaje de moscovita.	104
Figura 83: Simulación del coeficiente $p3$ en función del porcentaje de moscovita.	105
Figura 84: Simulación del coeficiente $p1$ en función de $D2$.	106
Figura 85: Simulación del coeficiente $p1$ en función de $D2$.	106
Figura 86: Simulación del coeficiente $p1$ en función de $D2$.	107
Figura 87: Simulación de la concentración crítica ϕc en función del porcentaje de caolinita.	108
Figura 88: Simulación de la concentración crítica ϕc en función del porcentaje de clorita.	109
Figura 89: Simulación de la concentración crítica ϕc en función del porcentaje de esmectita.	110
Figura 90: Simulación de la concentración crítica ϕc en función del porcentaje de moscovita.	111
Figura 91: Simulación de la concentración crítica ϕc en función de $D2$.	111
Figura 92: Valores obtenidos para cada modelo. Pre-exponencial de la función batch de Kynch.	126
Figura 93: Valores obtenidos para cada modelo. Exponente de la función batch de Kynch.	127
Figura 94: Valores obtenidos para cada modelo. Pre-exponencial de la función estrés efectivo del sólido.	127
Figura 95: Valores obtenidos para cada modelo. Exponente de la función estrés efectivo del sólido.	128
Figura 96: Valores obtenidos para cada modelo. Coeficiente cuadrático de la función de dosificación.	128

Figura 97: Valores obtenidos para cada modelo. Coeficiente lineal de la función de dosificación.	129
Figura 98: Valores obtenidos para cada modelo. Coeficiente libre de la función de dosificación.	129
Figura 99: Valores obtenidos para cada modelo. Concentración crítica.....	130
Figura 100: Simulación de la función batch de Kynch para la muestra R01.....	131
Figura 101: Simulación de la función estrés efectivo del sólido para la muestra R01. .	132
Figura 102: Simulación de la función de dosificación para la muestra R01.....	133
Figura 103: Simulación de la concentración crítica para la muestra R01.....	134
Figura 104: Simulación de la función batch de Kynch para la muestra R02.....	135
Figura 105: Simulación de la función estrés efectivo del sólido para la muestra R02. .	136
Figura 106: Simulación de la función de dosificación para la muestra R02.....	137
Figura 107: Simulación de la concentración crítica ϕc para la muestra R02.....	138
Figura 108: Simulación de la función batch de Kynch para la muestra R03.....	139
Figura 109: Simulación de la función estrés efectivo del sólido para la muestra R03. .	140
Figura 110: Simulación de la función de dosificación para la muestra R03.....	141
Figura 111: Simulación de la concentración crítica ϕc para la muestra R03.....	142
Figura 112: Simulación de la función batch de Kynch para la muestra R04.....	143
Figura 113: Simulación de la función estrés efectivo del sólido para la muestra R04. .	144
Figura 114: Simulación de la función de dosificación para la muestra R04.....	145
Figura 115: Simulación de la concentración crítica para la muestra R04.....	145
Figura 116: Simulación de la función batch de Kynch para la muestra R05.....	146
Figura 117: Simulación de la función estrés efectivo del sólido para la muestra R05. .	147
Figura 118: Simulación de la función de dosificación para la muestra R05.....	148
Figura 119: Simulación de la concentración crítica para la muestra R05.....	149
Figura 120: Simulación de la función batch de Kynch para la muestra R06.....	150
Figura 121: Simulación de la función estrés efectivo del sólido para la muestra R06. .	151
Figura 122: Simulación de la función de dosificación para la muestra R06.....	152
Figura 123: Simulación de la concentración crítica para la muestra R06.....	153
Figura 124: Simulación de la función batch de Kynch para la muestra R07.....	154
Figura 125: Simulación de la función estrés efectivo del sólido para la muestra R07. .	155
Figura 126: Simulación de la función de dosificación para la muestra R07.....	156
Figura 127: Simulación de la concentración crítica para la muestra R07.....	157
Figura 128: Simulación de la función batch de Kynch para la muestra R08.....	158
Figura 129: Simulación de la función estrés efectivo del sólido para la muestra R08. .	159
Figura 130: Simulación de la función de dosificación para la muestra R08.....	160
Figura 131: Simulación de la concentración crítica para la muestra R08.....	161
Figura 132: Simulación de la función batch de Kynch para la muestra R09.....	161

Figura 133: Simulación de la función estrés efectivo del sólido para la muestra R09. .	162
Figura 134: Simulación de la función de dosificación para la muestra R09.....	163
Figura 135: Simulación de la concentración crítica para la muestra R09.....	164
Figura 136: Simulación de la función batch de Kynch para la muestra R10.....	165
Figura 137: Simulación de la función estrés efectivo del sólido para la muestra R10. .	166
Figura 138: Simulación de la función de dosificación para la muestra R10.....	167
Figura 139: Simulación de la concentración crítica para la muestra R10.....	168
Figura 140: Simulación de la función batch de Kynch para la muestra R11.....	168
Figura 141: Simulación de la función estrés efectivo del sólido para la muestra R11. .	169
Figura 142: Simulación de la función de dosificación para la muestra R11.....	170
Figura 143: Simulación de la concentración crítica para la muestra R11.....	171
Figura 144: Simulación de la función batch de Kynch para la muestra R12.....	172
Figura 145: Simulación de la función estrés efectivo del sólido para la muestra R12. .	173
Figura 146: Simulación de la función de dosificación para la muestra R12.....	174
Figura 147: Simulación de la concentración crítica ϕ_c para la muestra R12.	175
Figura 148: Simulación de la función batch de Kynch para la muestra R13.....	175
Figura 149: Simulación de la función estrés efectivo del sólido para la muestra R13. .	176
Figura 150: Simulación de la función de dosificación para la muestra R13.....	177
Figura 151: Simulación de la concentración crítica para la muestra R13.....	178
Figura 152: Simulación de la función batch de Kynch para la muestra R14.....	179
Figura 153: Simulación de la función estrés efectivo del sólido para la muestra R14. .	180
Figura 154: Simulación de la función de dosificación para la muestra R14.....	181
Figura 155: Simulación de la concentración crítica para la muestra R14.....	182
Figura 156: Simulación de la función batch de Kynch para la muestra R15.....	183
Figura 157: Simulación de la función estrés efectivo del sólido para la muestra R15. .	184
Figura 158: Simulación de la función de dosificación para la muestra R15.....	185
Figura 159: Simulación de la concentración crítica para la muestra R15.....	186

Nomenclatura

Alfabeto latino

Variable	Representa
A	Integral respecto α de función de difusión
f_{bk}	Función batch de Kynch
g	Aceleración de gravedad
L	Punto de alimentación
Q_R	Flujo volumétrico en el estanque
t	Tiempo
S	Área transversal de la sección
x_1	Porcentaje de caolinita en la muestra
x_2	Porcentaje de clorita en la muestra
x_3	Porcentaje de esmectita en la muestra
x_4	Porcentaje de moscovita en la muestra
x_5	Porcentaje de sólido captado en un tamiz de 2 micras d_2

Alfabeto griego

Variable	Representa
α	Función de difusión
κ	Función de dosificación
$\Delta\rho$	Diferencia entre densidad del sólido y la densidad del agua
σ'_e	Derivada de la función Estrés Efectivo de sólido
ϕ	Concentración volumétrica
ϕ_0	Concentración volumétrica en el tiempo inicial
ϕ_L	Concentración volumétrica en la alimentación
ω_i	Peso de la i -ésima neurona de una red neuronal artificial

1. Introducción

1.1 Antecedentes Generales

En la actualidad muchas empresas, han comenzado a incorporar en sus procesos y sistemas algoritmos provenientes del área de Inteligencia Artificial, para así poder automatizar operaciones sencillas y diseñar modelos que predigan el comportamiento de alguna de sus partes, como por ejemplo, el área financiera de una empresa [1], la distribución de energía eléctrica en una ciudad [2], o la interacción de usuario en los teléfonos inteligentes [3]. Por esto varias mineras han invertido en el desarrollo e implementación de distintos simuladores para sus operaciones y procesos como herramientas de control y operación [1].

Dentro de estas operaciones se encuentra el tratamiento de relaves en toda faena minera basada en hidrometalurgia, y por tanto es muy importante simular de buena manera este sistema, ya que con él se conoce la cantidad de agua que se puede recircular al proceso y cuánto desecho sólido debe tratar la planta. Es recomendable tener buenas predicciones del espesor de relaves para conocer el tiempo con que sedimenta cada muestra, y así minimizar los tiempos de operación y ocupar de mejor manera los recursos económicos y humanos [4]. En un espesador se tratan relaves con un porcentaje de sólido entre el 20% y el 30%, obteniendo 2 salidas: un flujo de agua recuperada y un flujo con una concentración superior a 50% de sólidos. Esto además presenta varias ventajas como minimizar los riesgos de licuefacción, el volumen de almacenamiento de relaves y las posibilidades de filtraciones en el subsuelo [5].

Por otro lado, el uso de algoritmos de Machine Learning ha tenido un aumento considerable en el último tiempo. Algunos de los casos más destacados de implementaciones de algoritmos de Inteligencia Artificial son en camiones y vehículos para que puedan desplazarse adecuadamente sin la necesidad de un conductor [6], junto a su actual uso en aplicaciones clínicas en la predicción de accidentes cardíacos [7], y también en temas

contingente como la pandemia Covid-19 [8]. Sin embargo, aún existe mucha desconfianza con respecto al tema, por lo que las empresas están abiertas a invertir en esta tecnología, pero antes prefieren ver resultados concretos, puesto que la implementación de estos algoritmos implica una re-estructuración interna específica de los sistemas de control [9].

1.2 Motivación

Una empresa minera operativa en el país, que por motivos de confidencialidad no se indica su nombre, pretende desarrollar un sistema de simulaciones de todas las operaciones presentes en el proceso de extracción de cobre. Es por esto que se pide el diseño de un simulador predictivo de su espesador de relaves. Dicho trabajo es desarrollado a partir de un modelo fenomenológico en una dimensión espacial proveniente de la teoría de sedimentación, el cálculo de las funciones constitutivas para cada material, y el uso de métodos de diferencias finitas para su resolución numérica [10].

Como se mencionó anteriormente, las empresas mineras están dispuestas a invertir en esta nueva tecnología, sin embargo, requieren más información sobre los principales productos obtenidos (modelos, algoritmos, toma de decisiones, etc.) y sobre la calidad de éstos. De esta manera, es necesario estudiar el funcionamiento de este tipo de algoritmos en la simulación de procesos mineros, comparando el rendimiento de distintos enfoques de Machine Learning, ya que a nivel industrial el desarrollo de Machine Learning aún está en etapas tempranas, por lo que se pretende hacer una búsqueda exploratoria de distintas herramientas, para ver sus pros y contras en cada aplicación.

Sin embargo, el mayor riesgo de estas tecnologías es no implementarlas de manera temprana, ya que mientras más demoren las empresas en adoptarlas, mayor será la brecha en todos los aspectos con las empresas que sí lo hagan. Implementar algoritmos de Machine Learning es un proceso complejo y largo, por lo que *“Es probable que la IA genere ganadores y perdedores, y quienes comiencen a adoptar la tecnología pronto tendrán una ventaja significativa”* como dice Nigel Duffy, Líder de Inteligencia Artificial de Innovación de Ernst Young Global Limited [11].

1.3 Descripción del proyecto

El modelo utilizado para poder representar el proceso de sedimentación dentro del espesador de relaves corresponde a una simplificación que no incluye las variaciones en la viscosidad y se trabaja solo en una dimensión espacial (Altura z) y el tiempo (t). Esta expresión permite una representación del espesador de relave como se ilustra en la Figura 1, teniendo en cuenta la geometría del equipo y las zonas de clarificación, sedimentación y compactación.

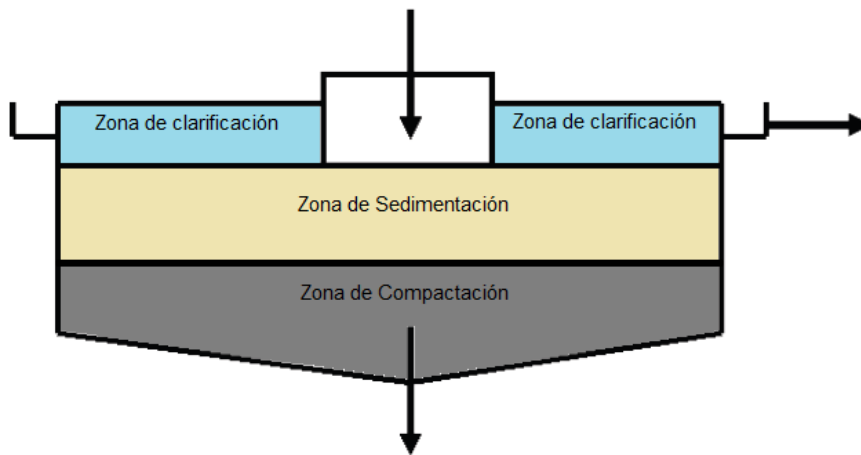


Figura 1: Representación gráfica del equipo simulado.

Por otro lado, se tienen datos experimentales del comportamiento de 15 materiales distintos, a los cuales para caracterizarlos se tienen las siguientes expresiones, que dan cuenta de propiedades esenciales del material para este proceso:

Tabla 1: Funciones constitutivas de estudio.

Variable	Expresión
Función batch de Kynch	$f_{bk}(\phi) = A\phi(1 - \phi)^b$
Estrés efectivo del sólido	$\sigma'_e(\phi) = Ce^{d\phi}$
Efecto de la dosis de floculante	$\kappa(\phi) = p_1\phi^2 + p_2\phi + p_3$

Estos parámetros (en negrita) se obtienen empíricamente a partir de pruebas experimentales de sedimentación. Se obtienen estos parámetros para caracterizar al sistema con las ecuaciones mostradas en la Tabla 1 [12].

La propuesta de la tesis es determinar rutas de búsqueda para dichos parámetros, utilizando algoritmos de Machine Learning para así disponer de una metodología que permita simular los procesos de sedimentación y predecir el comportamiento de distintos relaves en esta etapa. De esta manera, se pretende trabajar con redes neuronales artificiales (2 tipos de ellas), junto a un algoritmo de programación genética. Además, se propone realizar una validación exhaustiva, reportando el rendimiento de la herramienta para cada uno de los casos. Así, se proponen varios criterios para poder evaluar un algoritmo en específico (tiempo de simulación, costo computacional, poder de predicción, tendencia de variables, etc.). El alcance del trabajo implica un estudio sobre el desempeño de distintos algoritmos del área de Inteligencia Artificial en este tipo de procesos, pudiendo extender dicha metodología en diversas simulaciones similares, y a partir de esto generar una comparación para escoger algún tipo de algoritmo para este sistema o rubro.

1.4 Justificación

Una de las principales preocupaciones para las plantas mineras es lograr disminuir su consumo de agua, debido a la crisis medioambiental presente en la actualidad. Es por esto que los relaves, los cuales poseen un porcentaje de agua promedio de 75% [5], presentan una oportunidad para recuperar parte de ese porcentaje, recirculando el agua a los procesos anteriores, de esta manera disminuir el consumo de agua total de la planta.

Las proyecciones del consumo de agua por parte de las mineras en Chile, en el período entre 2018 y 2029, según la Comisión Chilena del Cobre (COCHILCO), presentan que para el año 2029 se tendrá un aumento de 12% en el consumo de agua continental, y un aumento de 230% en el consumo de agua de mar [13]. El espesamiento de relaves es, por tanto, una operación que presenta grandes beneficios, tanto económicos como ambientales a las empresas mineras. Sin embargo, para que ocurra la sedimentación en el espesador de relaves de manera efectiva, se necesita de

ciertas propiedades constitutivas de los materiales que sedimentan. Esto quiere decir que existirán materiales que demoren tiempos que exceden los tiempos requeridos por la buena operación de la planta, así como también existirán materiales que no logren sedimentar de buena manera, dejando partículas en suspensión, pero que al agregar una cierta dosis de floculante se beneficie la sedimentación.

Para la obtención de las funciones constitutivas, las empresas no tienen modelos matemáticos, sino que han optado por llevar las muestras a laboratorios para calcular los valores empíricamente. Sin embargo, muchas de las muestras pueden comportarse de manera tal de tener mediciones que exceden las posibilidades del laboratorio, por lo que la obtención teórica de estos parámetros permitiría a las empresas contar con una aproximación de éstos. Así mismo, el algoritmo le podría servir a los laboratorios para entregar resultados aproximados en un menor tiempo.

2. Objetivos y limitaciones

2.1 Objetivos

Objetivo General:

- Estudiar el desempeño de distintos algoritmos del área de Inteligencia Artificial, aplicado a un sistema de espesamiento de relaves, en la obtención de los parámetros de las ecuaciones utilizadas para la modelación fenomenológica de dicho proceso.

Objetivos específicos:

- Estudiar el fenómeno del espesamiento y las características del sistema de espesadores de relaves.
- Implementar distintos algoritmos de Machine Learning, para la obtención de parámetros de funciones constitutivas en un simulador predictivo de espesadores de relaves.
- Validar y analizar resultados obtenidos a partir de los resultados experimentales de muestras de relaves.

2.2 Limitaciones

La teoría de la sedimentación presenta modelos fenomenológicos más avanzados, que no permiten una solución mediante métodos numéricos, debido principalmente a la complejidad del modelo y la capacidad actual de procesamiento [13,14], por lo que se debe trabajar con un modelo simplificado, como el que se presenta en la sección 3.3. Por otro lado, la planta no cuenta con todos los sensores requeridos para validar los datos de la simulación, vale decir, sensores de viscosidad y de nivel de la altura de la zona de compresión, y que no son necesarios para la operación. Sin embargo, con los equipos disponibles, se conoce la alimentación total al sistema, la concentración y el flujo de salida de cada uno de los equipos.

Finalmente, al tratarse de algoritmos de Machine Learning que serán entrenados con los datos propios de la minera, el funcionamiento adecuado de esta herramienta solo se puede garantizar dentro de rangos similares para todas las condiciones de operación de la planta. Los modelos obtenidos a partir de las redes neuronales y la programación genética no pueden ser generalizados para extrapolarlos en otros sistemas. Sin embargo, la metodología planteada en este trabajo puede ser replicable en otros tipos de problemas que requieran el uso de este tipo de algoritmos, tanto dentro de los procesos mineros como en la industria química en general.

3. Marco teórico

3.1 Proceso del concentrado de cobre y molibdeno

El proceso de concentrado de cobre y molibdeno se divide en 3 procesos principales: extracción, conminución y concentración, los cuales se subdividen en distintas operaciones unitarias involucradas en el proceso de los espesadores de relave [14], como se observa en la Figura 2.

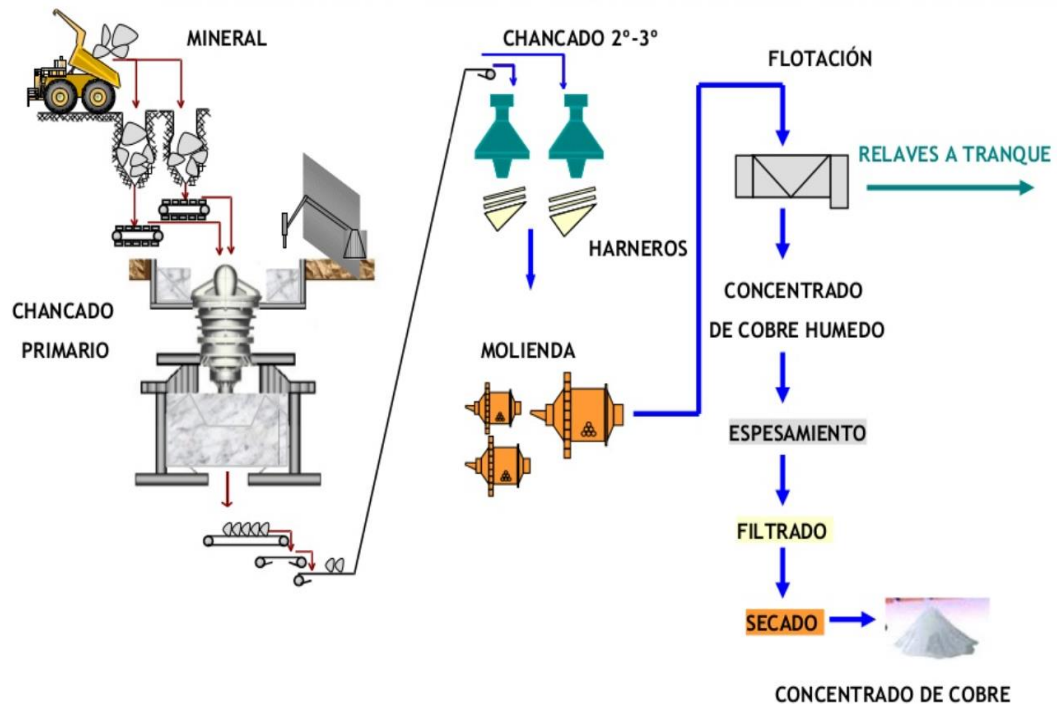


Figura 2: Proceso del cobre. Modificado de [10].

Extracción

Su objetivo es sacar la porción de mineral desde la mina, ya sea subterránea o tajo abierto. Está definido por el tipo de roca, las leyes del mineral y sus subproductos, junto a los costos asociados [15]. Se compone de 4 operaciones unitarias [14]:

- Perforación y tronadura: Consiste en hacer una cavidad dentro de la roca para soportar los explosivos involucrados. Esta operación representa aproximadamente del 10% al 15% del costo total de la operación.
- Carguío y transporte: Consiste en recoger el material para cargarlo en camiones de gran tonelaje, con ayuda de palas hidráulicas, eléctricas y cargadores frontales, representando entre un 10% y un 20% del costo de operación minera.

Conminución

El objetivo de la conminución es disminuir el tamaño del material obtenido en la extracción, exponiendo el mineral de interés para que se pueda recuperar en los siguientes procesos [14]. Se divide en 2 operaciones:

- Chancado: Aquí se trituran las rocas hasta alcanzar tamaños de aproximadamente media pulgada [10]. Este proceso se debe realizar en varios circuitos debido a que aún no se desarrolla una tecnología que permita hacerlo en una sola etapa.
- Molienda: Se produce una mezcla entre el material del chancado junto con agua para formar una pulpa. Su propósito es reducir aún más el tamaño del material al nivel de micrones [16].

Concentración

En este proceso se separan los minerales con valor comercial de los demás componentes de la pulpa, retirando el mayor porcentaje de agua [16]. Su operación principal es la flotación, la cual consiste en capturar los minerales hidrofóbicos de la pulpa mediante burbujas de aire [17].

Espesamiento

Esta es una de las etapas claves en muchos procesos hidrometalúrgicos. Corresponde a una operación de separación sólido-líquido, cuyo principio consiste en el agregamiento de partículas finas de una suspensión mediante

un floculante, aumentando su tamaño y permitiendo que ellas sedimenten efecto de la gravedad, produciendo una solución clara y un sedimento denso [18].

Filtración y secado

Proceso de separación sólido-fluido mediante el cual el sólido es separado del fluido en una suspensión a través de un lecho poroso, denominado filtrante, formando una torta o queque de filtración [18]. El agua recuperada por el filtrado y el agua de lavado de membranas es colectado en un pozo, donde retorna a la alimentación del espesador de relaves. Luego del filtrado, el concentrado tiene un 90% de sólidos, y un porcentaje de humedad del 9%. El secado consiste en depositar los restos sólidos en piscinas de evaporación [19].

3.2 Caracterización del espesador de relaves estudiado

Debido a la continua explotación de los yacimientos, se ha observado una disminución de la ley de cobre a nivel mundial, por lo que cada vez se extraen mayores masas de material para mantener la producción [16]. Esto implica costos de operación más elevados y un aumento en la cantidad de relaves producidos [15]. En los espesadores de relaves, la pulpa se separa en agua y los desechos del proceso. Esto ocurre debido a que existe una gran diferencia de densidad entre el sólido y el líquido [20], lo que provoca que las partículas sean retiradas por el *underflow* mientras que el agua sale libre de la mayor parte de los sólidos por el *overflow* [21].

El material fino presenta una velocidad de sedimentación excesivamente baja, por lo que presentan un problema para la operación de los espesadores. Esto puede ser, en parte, solucionado con el uso de floculantes que logran aglomerar dichas partículas [22,23]. El rendimiento del floculante depende del pH y de la conductividad, lo cual distintos tipos de floculantes se comportan de diferentes formas respecto a sus condiciones de operación, pudiendo dejar de funcionar en caso de pH muy ácido o elevada conductividad [22].

Existen distintos tipos de espesadores, en primer lugar, se tiene los espesadores convencionales, que no utilizan floculante y ni cuentan con un sistema de dilución para la alimentación, llegando a porcentajes de sólido de entre 40 a 60% p/p [23]. En segundo lugar, están los espesadores *high rate*, que utilizan floculantes y alimentación con dilución. También existen los espesadores *high density* que se utiliza para recuperar una mayor cantidad de agua en las grandes faenas que entraron en operación con bajas leyes de mineral [24]. Finalmente, los espesadores de pasta, que tienen una altura lateral mucho mayor que los demás tipos de espesadores y su producto es un relave con concentración máxima de sólidos conocido como pasta [23].

El espesador de relaves utilizado por la empresa corresponde a un espesador *high rate* semi-cónico de columna central, con 2 pendientes distintas, por lo que se requiere implementar un modelo que considere la variación de área dentro del equipo. El espesador cuenta con una altura máxima de 16,51[m] y un diámetro máximo de 99[m]. En la Figura 3 se muestra un diagrama especificando las dimensiones más importantes del espesador.

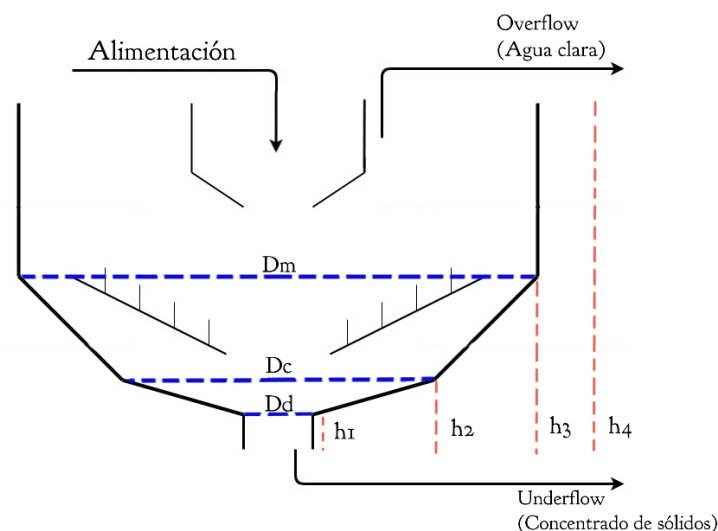


Figura 3: Diagrama de la geometría del espesador de estudio.

En la Tabla 2 se presenta las características de diseño y operación del espesador de la planta.

Tabla 2: Caracterización del espesador de relaves estudiado.

Tipo	High Rate de columna central
Sistema de dilución	Autodilución y dilución forzada
Concentración de sólido Alimentación	24 – 25%
Concentración de sólido Feedwell	7 – 14%
Concentración de sólido Descarga	50 – 52%
Dosificación de floculante	13 – 15 [g/t]
Flujo Alimentación	6900 – 8000 [m ³ /h]
Densidad Alimentación	1,18 [t/m ³]
Densidad Descarga	1,47 [t/m ³]

3.3 Modelo de espesador de relaves

Los modelos matemáticos utilizados para describir el fenómeno ocurrido en los espesadores de relaves son los basados en la teoría de la sedimentación y los modelos de fluido dinámica computacional (CFD). El modelo estudiado está basado en la teoría de la sedimentación, y debido a la complejidad que presenta el modelo fenomenológico completo, se deben realizar simplificaciones a éste. Los supuestos tomados para ello son los siguientes:

Tabla 3: Supuestos utilizados para el diseño del modelo.

Supuestos	
(1)	El sólido y el fluido son dos medios continuos superpuestos
(2)	No se considera el efecto de las paredes del equipo
(3)	Todas las variables dependen exclusivamente de la altura y el tiempo
(4)	Existe un punto ($z = L$) de alimentación y un punto de salida ($z = 0$)
(5)	La viscosidad es despreciable

Estos supuestos implican que el modelo no puede describir lo que ocurre en la zona de clarificación, además de no tener solución analítica, por lo que es necesaria la implementación de métodos numéricos [12,25,26]. El modelo considera la sedimentación gravitacional, la sedimentación por convección

y la consolidación de material en área variable, como se observa en la ecuación (3.3.1). Incluye el efecto de la modificación de la dosis de floculante y los fenómenos de sedimentación convectiva y compactación. En la ecuación (3.3.2) se muestra la condición inicial del espesador, en la ecuación (3.3.3) la condición de alimentación, en la ecuación (3.3.4) la condición de borde en la descarga, y finalmente la ecuación (3.3.5) representa el balance masa del sistema.

$$S(z) \frac{\partial \phi}{\partial t} + \frac{\partial}{\partial z} (Q_R \phi + S(z) K f_{bk}(\phi)) = \frac{\partial}{\partial z} \left(S(z) K \frac{\partial A(\phi)}{\partial z} \right) \quad (3.3.1)$$

$$\phi(z, 0) = \phi_0, \quad z \in (0, L) \quad (3.3.2)$$

$$f_{bk}(\phi) - \alpha(\phi) \frac{\delta \phi}{\delta z} \Big|_{z=0} = 0 \quad (3.3.3)$$

$$\phi(L, t) = \phi_L, \quad t \in (0, T) \quad (3.3.4)$$

$$\frac{\partial A(\phi)}{\partial z} = \alpha(\phi) = \frac{f_{bk} \sigma'_e}{\Delta \rho g} \quad (3.3.5)$$

El modelo presentado en la ecuación (3.3.1) es del tipo hiperbólico-parabólico, y corresponde a una simplificación utilizada exitosamente para observar el comportamiento de perfiles de concentraciones en el tiempo, teniendo un buen rendimiento en sistemas de control [25]. La principal limitación de este modelo es que requiere de sensores especializados para caracterizar el perfil de concentración dentro del equipo [25], los cuales estaban presentes en la medición de los datos experimentales de esta memoria.

En las Figura 4, 5 y 6 se muestran algunos ejemplos de perfiles de concentración de sólido en el espesador. Estos gráficos corresponden a la concentración de sólidos en función de la altura del espesador con un tiempo de simulación de 4 horas, que corresponde al tiempo que la empresa utiliza para la sedimentación de las muestras. Cada una de las muestras presentan un perfil de sedimentación distinto, dependiendo de la composición de arcillas y las propiedades granulométricas de los diferentes

relaves que se simulan. Como ejemplo, se puede observar que en la muestra R03 la concentración disminuye drásticamente al aumentar la altura, en especial se aprecia un cambio de concentración abrupto a 1 [m] de altura del espesador, mientras que en las otras 2 muestras esta diferencia no es tan notoria. Esto se debe a que la muestra R03 posee un menor porcentaje de clorita y esmectita, mientras que posee un mayor porcentaje de caolinita, en comparación a las muestras R01 y R04. La caolinita es la arcilla que responde de mejor manera a la adición de floculante, por lo que se aglomera y logra sedimentar, mientras que para la clorita y la esmectita el floculante es menos efectivo.

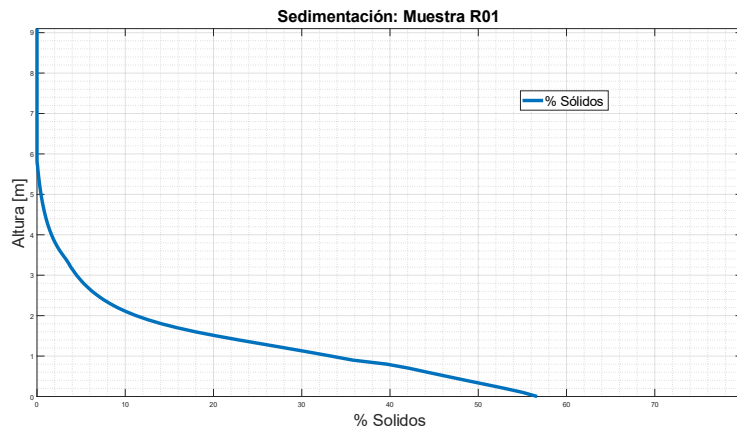


Figura 4: Perfil de concentración de sólidos para la muestra R01.

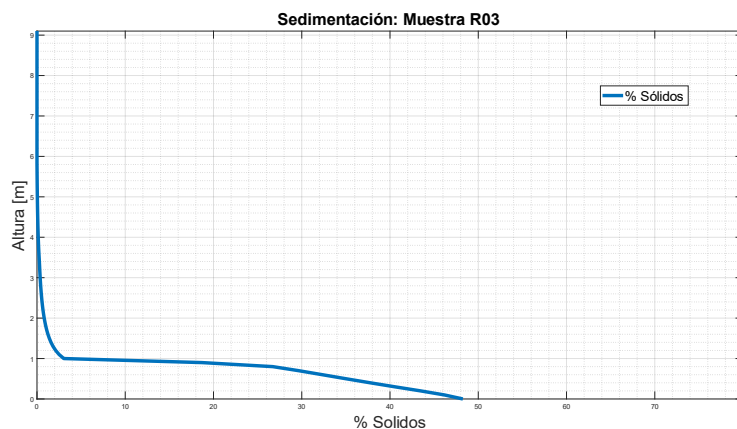


Figura 5: Perfil de concentración de sólidos para la muestra R03.

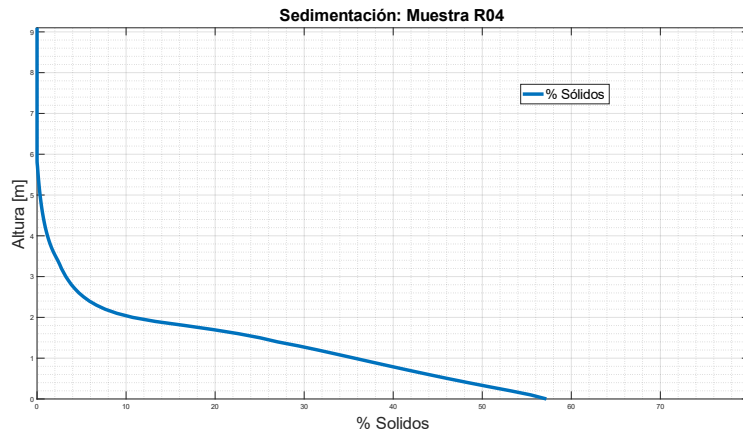


Figura 6: Perfil de concentración de sólidos para la muestra R04.

3.4 Definición de funciones constitutivas

Dentro de la teoría de la sedimentación, existen funciones las cuales se utilizan para representar los distintos fenómenos que ocurren en el sistema de estudio. A estas expresiones se les llama funciones constitutivas, y representan la convección, la sedimentación y la consolidación en el equipo. A continuación, se describen las funciones constitutivas utilizadas para el estudio de espesadores de relaves:

- **Función batch de Kynch (f_{bk}):** Representa la densidad de flujo volumétrico y, por tanto, corresponde a la sedimentación gravitacional en el proceso. Se obtiene a partir de las pruebas de sedimentación estática, variando la dilución inicial [12]. En los trabajos de Becker [26], a partir de sus estudios con muestras de relaves de la División Chuquicamata de Codelco-Chile, se propone la siguiente expresión:

$$f_{bk} = A\phi(1 - \phi)^b \quad (3.4.1)$$

Donde $b > 0$, y $A < 0$ es una propiedad exclusiva de cada material y representa la velocidad de la solución de un solo flóculo en el líquido puro, en [m/s] [27].

- **Función de estrés efectivo del sólido** (σ_e): Corresponde al fenómeno de la consolidación, la cual comienza a operar una vez que se haya logrado la concentración crítica. Se obtiene a partir del perfil de concentración final de compactación dinámica. Al igual que el caso anterior, Becker propone la siguiente expresión matemática [26]:

$$\sigma_e = C e^{d\phi} \quad (3.4.2)$$

Esta función representa el traspaso progresivo de la fuerza del flujo del agua, entre los intersticios de la formación del sedimento al esqueleto del sólido, esta fuerza sale de entre el sedimento hacia fuera de él [29].

- **Función de dosificación de floculante** (κ): Corresponde a la cantidad óptima de floculante para una cierta concentración de la muestra. Se obtiene a partir de las pruebas de sedimentación estática, variando la dosis de floculante, y tiene una forma polinómica de segundo grado.

$$\kappa = p_1\phi^2 + p_2\phi + p_3 \quad (3.4.3)$$

- **Concentración crítica** (ϕ_c): Representa la concentración donde las partículas comienzan a tener contacto efectivo entre sí, permitiendo el transporte difusivo producto de la consolidación. Se obtiene a partir de las pruebas de sedimentación estática o compactación dinámica. Corresponde a una constante numérica.

$$\phi_c(\phi) = \phi_c \quad (3.4.4)$$

3.5 Efecto de arcillas en las funciones constitutivas

En cuanto a los factores determinantes de la reología de las muestras de relaves, estudios realizados por varios autores afirman que tanto la viscosidad como el límite de fluencia tienden a aumentar con la

concentración y granulometría fina de una suspensión [28]. Con respecto a la viscosidad, a medida que aumenta el espacio ocupado por sólidos, existe mayor desviación de las líneas de flujo del medio dispersante, aumentando las fuerzas viscosas de la suspensión. Mientras que, si las partículas son muy pequeñas, acciones viscosas producen aglomeración de éstas, aumentando la viscosidad y la probabilidad de colisión entre partículas de sólidos y, por lo tanto, habrá una mayor oposición a las partículas que intentan moverse a la fase dispersante [29].

Las arcillas son aluminosilicatos hidratados, con distintas cantidades de hierro, magnesio, metales alcalinos y otros cationes. Se caracterizan por estar formadas por partículas muy finas (menor a $2\mu m$) y de forma aplanada. Con esto, las arcillas, en menor o mayor medida, comparten las siguientes características [28]:

- Plasticidad
- Propiedades coloidales
- Propiedades absorbentes con respecto al agua y a cationes.

Según reportes de experimentos sobre la influencia de filosilicatos en muestras de relaves, se evidencia que a medida que aumenta el contenido de estas especies disminuye la velocidad de sedimentación de las muestras. La especie de caolinita es la que presenta el mayor efecto individual sobre la velocidad de sedimentación, disminuyéndola drásticamente [30]. Por otro lado, al aumentar el parámetro D_2 disminuye la velocidad de sedimentación, debido a que la muestra tiende a presentar un comportamiento más coloidal [30].

En cuanto al efecto de floculante, se registra en estudios realizados con relaves que la velocidad de sedimentación aumenta a medida que aumenta la dosis de éste. Se comprueba, en varios estudios, la efectividad de utilizar floculantes selectivamente los componentes de menor tamaño, a pesar de que la recuperación en peso se vea levemente afectada. En la Figura 7 se observa el comportamiento registrado para este análisis.

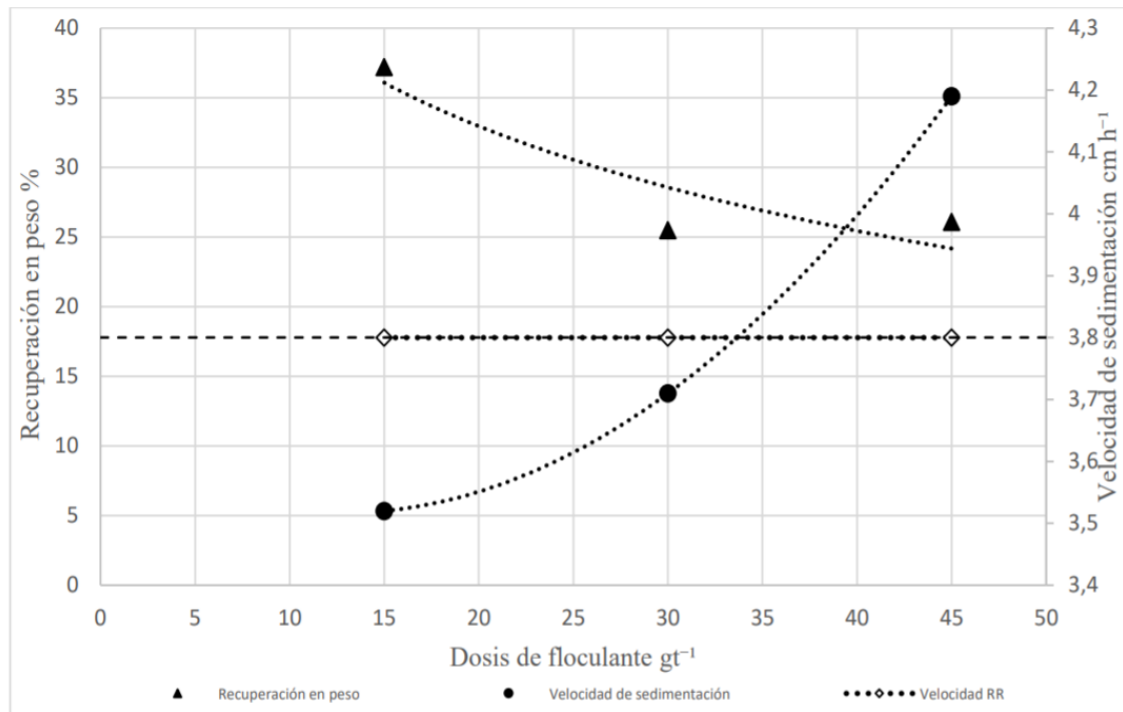


Figura 7: Efecto de la dosificación de floculante en la recuperación y velocidad de sedimentación [45].

Es por esto, que la presencia de estas arcillas en las muestras afecta notablemente la velocidad de sedimentación de los relaves y el nivel de compactación de éstos, siendo entonces, junto a la gravedad específica, variables que permiten caracterizar el comportamiento en la sedimentación de las muestras estudiadas. Así, a nivel general, a medida que aumentan las variables analizadas en este estudio (% de caolinita, % de clorita, % de esmectita, % de moscovita y D_2), se verá perjudicada la velocidad de sedimentación de la muestra (representada por la función batch de Kynch f_{bk}), así como también el nivel de compactación (representado por la función de estrés efectivo del sólido σ_e). Por otra parte, a medida que aumenta el porcentaje de estas arcillas en las muestras, se debe aumentar también la dosis de floculante para poder ponderar este efecto, como también aumentará el valor de la concentración crítica, debido a que las muestras les costará más entrar en la fase de compactación.

3.6 Inteligencia Artificial en procesos mineros

En la actualidad, las herramientas informáticas, como los algoritmos de Inteligencia Artificial han tenido un gran impacto en toda la industria minera, tanto en la faena como en su logística. Algunas de sus aplicaciones más relevantes se encuentran:

- Automatización de procesos [31].
- Estimación y predicción [32].
- Control y gestión de recursos entre procesos [1].

Actualmente, mediante modelos predictivos se obtienen los datos posibles que permiten incluso identificar fallas técnicas en un equipo [33]. Las tecnologías prioritarias en la minería comprenden la robótica y automatización, vehículos no tripulados, ciberseguridad, visualización de datos en tiempo real y análisis de datos, por lo que se denomina *Revolución Industrial 4.0* a la implementación de este tipo de tecnología, la cual significará un cambio importante en la forma de trabajo de numerosas empresas. Los principales beneficios de la implementación de estas tecnologías es un mejor rendimiento de equipos, ahorros administrativos/operativos y una toma de decisiones más eficiente [34].

3.7 Análisis exploratorio de datos

Diagrama de cajas y bigotes (box-and-whisker plots)

Una de las formas más fácil y poderosa para poder visualizar y analizar de manera exploratoria la dispersión de datos y variables dentro de un modelo corresponde a los diagramas de caja (*box plot*), también conocidos como diagramas de caja y bigotes (*box-and-whisker plots*), creado por el estadístico John Tukey [35]. Este tipo de diagrama utiliza parámetros estadísticos como la media, la mediana y el rango intercuartílico, los cuales son robustos en presencia de sesgos y valores atípicos, además de no requerir suposiciones a priori sobre la distribución de la población de datos. Con ello, representa el rango completo de los datos y proporciona información sobre las colas y distribución de la serie [36]. A continuación,

se definen las propiedades de un diagrama de cajas, como se observa en la Figura 8 [35].

- Divide los datos de la muestra en cuartiles: 2 paneles en forma de cajas y 2 áreas encerradas en llaves, además de los valores atípicos fuera de estas áreas.
- Los paneles o cuadros abarcan el 50% de los datos de la muestra. Los límites exteriores de dichos cuadros (llamados “bisagras”) indican el primer cuartil (donde se presenta el 25% de los datos de menor valor) en el límite inferior, y el tercer cuartil (donde se presenta el 25% de los datos de mayor valor).
- El límite entre los paneles corresponde a la mediana de los datos (segundo cuartil).
- La distancia entre los límites exteriores de los cuadros se denomina rango intercuartílico (IQR), la cual es una medida de dispersión similar a la desviación estándar. A medida que aumenta el IQR la dispersión de los datos será mayor.
- Las áreas en llaves (bigotes) muestran la extensión de rango de datos restantes. Comienzan en los límites exteriores de los paneles (primer y tercer cuartil) y terminan en los valores considerados extremos, que usualmente se considera una distancia de $1,5IQR$ más allá del primer y tercer cuartil.
- Los valores más allá de los extremos se denominan valores atípicos y se marcan como puntos individuales.

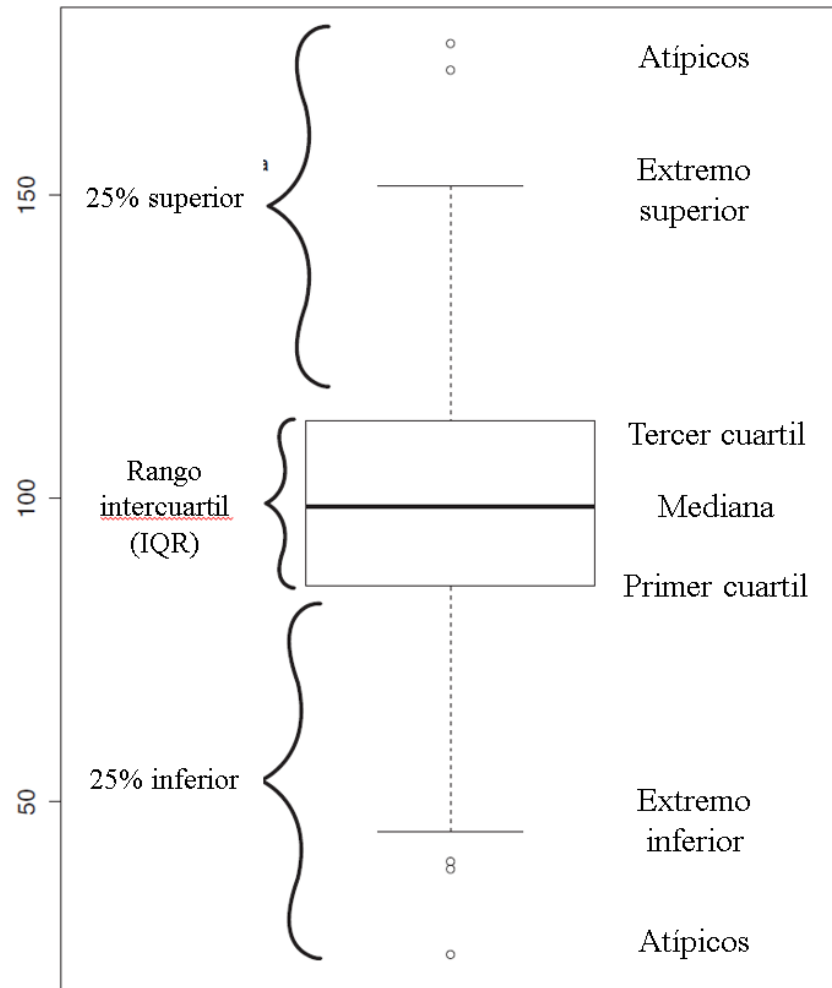


Figura 8: Representación de un diagrama de cajas y bigotes. Adaptado de [37].

A pesar de la basta información que logra representar los diagramas de cajas, éstos poseen desventajas, las cuales se enlistan a continuación:

- Deben usarse con una muestra de al menos 5 datos.
- No son apropiados para datos ordinales ni nominales.
- Las áreas en llaves no corresponden a barras de error estándar ni barras de desviación estándar.

3.8 Algoritmos de Machine Learning

Machine Learning corresponde a una rama de la Inteligencia Artificial, y consiste en crear sistemas que puedan aprender por sí solos. Esto es posible desarrollando una serie de operaciones automáticas que permitan identificar patrones en un set de datos, sin la intervención de un ser humano para ello. Esto puede significar una ventaja para cuando se tiene un gran volumen de datos o un tiempo corto para obtener resultados [37].

La clave de la capacidad de un sistema de Machine Learning se encuentra en el diseño e implementación de los árboles de decisiones en base a datos previamente conocidos por el sistema. El algoritmo se basa de experiencias y evidencias en forma de datos, con los que aprende y comprende las tendencias ocultas en éstos. A partir de un número de ejemplos de una situación, puede elaborar un modelo que puede deducir y generalizar un comportamiento observado [38].

Existen 3 tipos de algoritmos de Machine Learning: Aprendizaje supervisado, donde la información entregada al algoritmo es detallada y ya está catalogada; Aprendizaje no supervisado, donde no se entrega la información catalogada ni agregada de ninguna manera y se basan en la comprensión y abstracción de patrones; y Aprendizaje por refuerzo, donde el algoritmo aprende a partir de “*ensayo y error*” y a menudo un sistema de premio y castigo [34].

Programación Genética

La Programación Genética (GP) pertenece a la familia de algoritmos evolutivos, específicamente proviene de los Algoritmos Genéticos, en donde se hace un paralelismo biológico entre los individuos de una determinada población y las soluciones de un problema matemático [36]. En GP se crea una población de posibles soluciones al problema, específicamente, distintas expresiones matemáticas que deben representar a un set de datos experimentales [34]. A cada una de estas expresiones (las cuales están compuestas de operaciones elementales como suma, multiplicación, exponencial, funciones trigonométricas, logaritmo, etc.) se

le calcula un error, que corresponde a la diferencia entre el modelo y el set de datos experimentales [39]. De esta manera se puede observar qué individuos son más “aptos” para el problema específico. Al igual que la teoría de la Selección Natural, postulada por Charles Darwin, los individuos con un mayor índice de adaptabilidad (menor error) serán los que transmitan su “material genético”, que en este caso serían los términos que componen a la expresión [34]. Para esto se cuentan con operadores genéticos, como la mutación y el crossover (entrecruzamiento de genes), lo que producirá “hijos”(árboles binarios de la próxima iteración del algoritmo) que serán combinaciones de las expresiones matemáticas que había antes, por lo que, si este procedimiento se realiza varias veces, las expresiones deberán disminuir su error y encontrar fórmulas que representen de buena manera al sistema físico. En la Figura 9 se muestra un diagrama de flujo básico de la Programación Genética, con sus etapas y decisiones fundamentales.

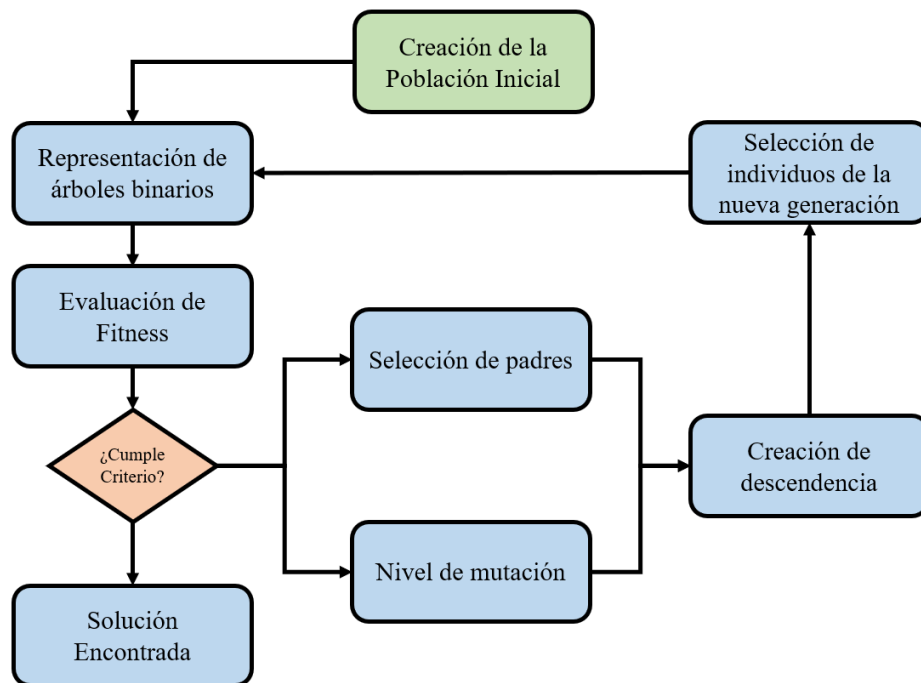


Figura 9: Diagrama de flujo del algoritmo de programación genética.

La estructura interna del algoritmo se basa en árboles binarios, la cual es una herramienta matemática para expresar información [40]. En este caso, los nodos intermedios del árbol corresponden a operaciones elementales y

los nodos terminales representan variables o constantes que pueda tener el modelo [41]. Dentro de los parámetros que se deben ajustar en la Programación Genética se encuentran: número de individuos (árboles) creados, tipo de selección de crossover, nivel de mutación, tipo de selección de los nuevos individuos, número de generaciones, entre otras [34].

Redes Neuronales Artificiales

Una red neuronal artificial (NN por sus siglas en inglés) es un modelo matemático inspirado en el comportamiento de las neuronas biológicas y su forma de organización, funcional y estructuralmente. Los componentes de la red (neuronas) funciona de manera paralela entre ellas, lo que hace que pueda procesar grandes volúmenes de información y encontrar tendencias en los datos, tal como hace un cerebro humano [41].

En esta estructura, las neuronas corresponden a puntos donde se toma una determinada decisión en el sistema, y la razón de que decida realizar una acción u otra está condicionado a si la neurona anterior logra enviarle una señal lo suficientemente intensa como para que se active la neurona [42]. En otras palabras, cada neurona debe recibir una señal mayor a un umbral establecido para que pueda activarse (dicha información se encuentra en la función de activación y el peso de cada una de las conexiones neuronales). En la Figura 10 se observa una estructura típica de una red neuronal, identificando los distintos tipos de neuronas existentes [42].

Las neuronas se organizan en capas [43], donde se pueden diferenciar 3 tipos:

- Capa de entrada: En este caso, las neuronas corresponden a lugares donde se almacenan los datos y variables de entrada al sistema, es decir, toda la información que se considere valiosa para el diseño y entrenamiento de la red (datos experimentales, reportes históricos, leyes fundamentales, etc.). Entregan la información a la primera capa oculta. Ver zona azul en la Figura 10.
- Capas ocultas: En esta capa cada neurona representa acciones en el sistema, por lo que dependerá de los valores entregados en la capa de

entrada cuáles neuronas se activarán y qué rutas serán viables. La última capa oculta se conecta con la capa de salida. Ver zona anaranjada en la Figura 10.

- Capa de salida: Las neuronas corresponden a estructuras que representan las variables de salida del sistema, es decir, los resultados arrojados por la red. Estos valores pueden corresponder a variables numéricas o booleanas (toma de decisiones). Ver zona verde en la Figura 10.

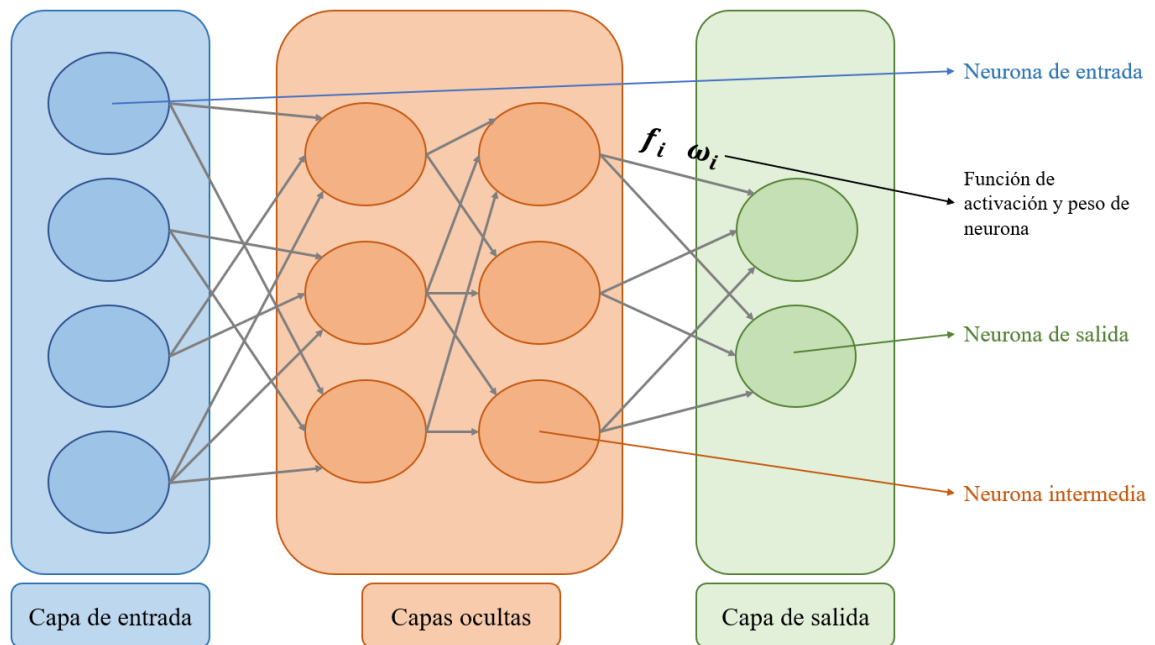


Figura 10: Diagrama de una red neuronal artificial. Adaptado de [43].

La función de activación ($f(x)$) está estrechamente ligada a la salida de la neurona, teniendo la siguiente forma [44]:

$$Y = f\left(\sum_{i=1}^n \omega_i x_i\right) \quad (4.5.1)$$

Donde Y es la salida (respuesta) de la neurona, ω_i el peso de la i -ésima neurona, x_i el estímulo (información) de la i -ésima neurona, y $f(\sum \omega_i x_i)$

corresponde a la función de activación evaluada sobre la combinación lineal de todos los estímulos que entran a la neurona. Dependiendo de la acción realizada por la neurona, se suele variar la función de activación. En el Anexo A se pueden observar las funciones de activación más comunes para el diseño de redes neuronales [44].

Redes neuronales recurrentes (RNN)

Comúnmente, las conexiones dentro de una red neuronal van una sola dirección (desde la capa de entrada hacia la capa de salida). Sin embargo, una red neuronal recurrente se caracteriza por poseer conexiones en el sentido contrario, lo cual corresponde en una retroalimentación adicional para las neuronas [45]. El ejemplo más básico es el de una neurona recurrente, como se observa en la Figura 11. La neurona entregará una respuesta y por cada estímulo x que reciba, como toda neurona artificial. Además, recoge información para sí misma y mejorar su desempeño, por lo que en una línea de tiempo se tiene que la misma neurona recibirá la información correspondiente a su instante de tiempo (t), e información de su salida anterior ($t - 1$) [45].

Así, una red neuronal recurrente se implementa tal que, en cada paso temporal, cada neurona recibe 2 entradas: la entrada de la capa anterior y la salida del tiempo anterior de la misma neurona.

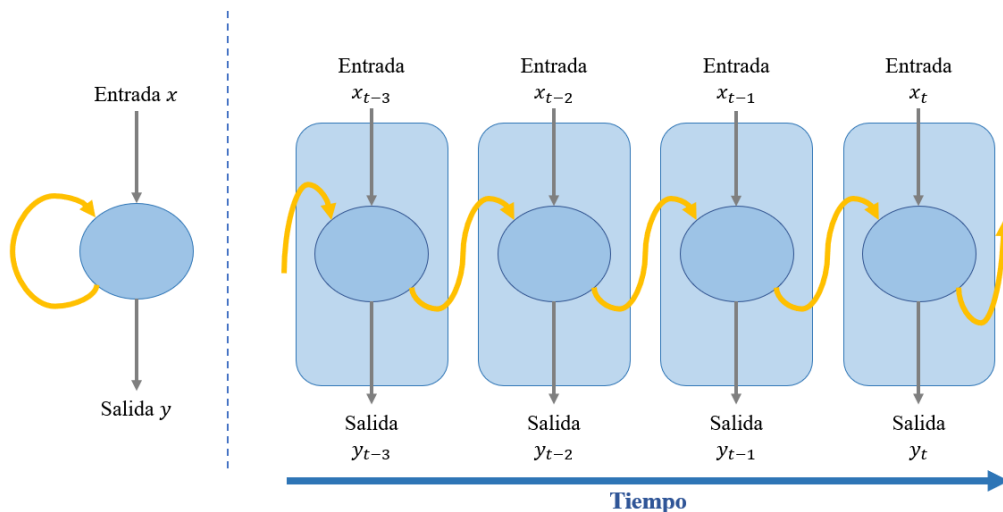


Figura 11: Diagrama de red neuronal recurrente. Adaptado de [46].

En este caso se tiene la siguiente fórmula para la salida de la neurona [46]:

$$Y_t = f \left(\sum_{i=1}^n \omega_{it} x_{it} + u_{it-1} y_{it} \right) \quad (4.5.2)$$

Donde y_{it-1} y u_{it-1} corresponden a la salida y el peso de la salida de la i -ésima neurona en el tiempo $(t - 1)$, respectivamente. Las ventajas de este tipo de red neuronal son [45]:

- Reconocimiento de secuencias: Se reproduce un patrón de salida particular cuando se especifica una secuencia de entrada.
- Reproducción de secuencias: La red debe ser capaz de generar el resto de una secuencia cuando ve una parte de ella.
- Asociación temporal: Una secuencia de salida particular se debe producir en respuesta a una secuencia de entrada específica.

Redes neuronales convolucionales (CNN)

En este tipo de red neuronal, cada parte de la red se entrena para realizar una tarea específica, por lo que el aprendizaje es más corto y rápido (menos neuronas aprenden una tarea más simple) y disminuyen el número de capas ocultas en comparación a una red neuronal convencional. Este tipo de red es muy potente, en especial en el análisis de imágenes (detectando de forma separada las líneas, bordes, colores, figuras, etc.) y en problemas donde se pueden identificar distintos subgrupos de elementos, direccionando el aprendizaje de la red [47].

La estructura típica de una red neuronal convolucional cuenta con múltiples capas ocultas, alternando entre capas convolucionales y capas de reducción, como se muestra en la Figura 12. En la convolución se realizan operaciones para generar un mapa de características, identificándolas y catalogando la información a través de un filtro, el cual es una propiedad intrínseca de la neurona. Por otro lado, la reducción disminuye la cantidad de parámetros ya que considera las características más comunes (máximo, promedio, etc.), pero a la vez perdiendo precisión en el proceso. En la Figura 13 se observan estas 2 operaciones con ejemplos pequeños [45].

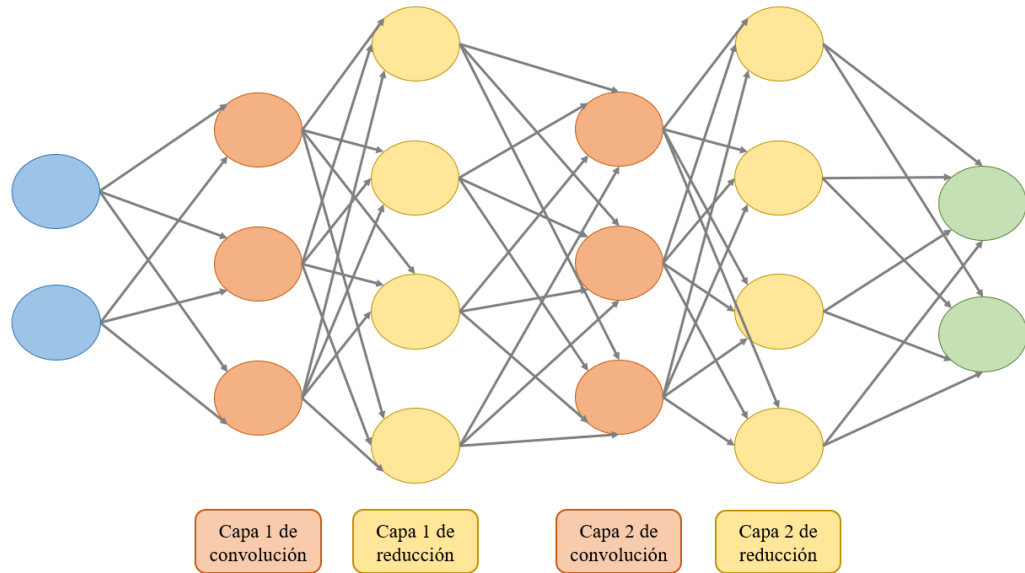


Figura 12: Diagrama de red neuronal convolucional. Adaptado de [47].

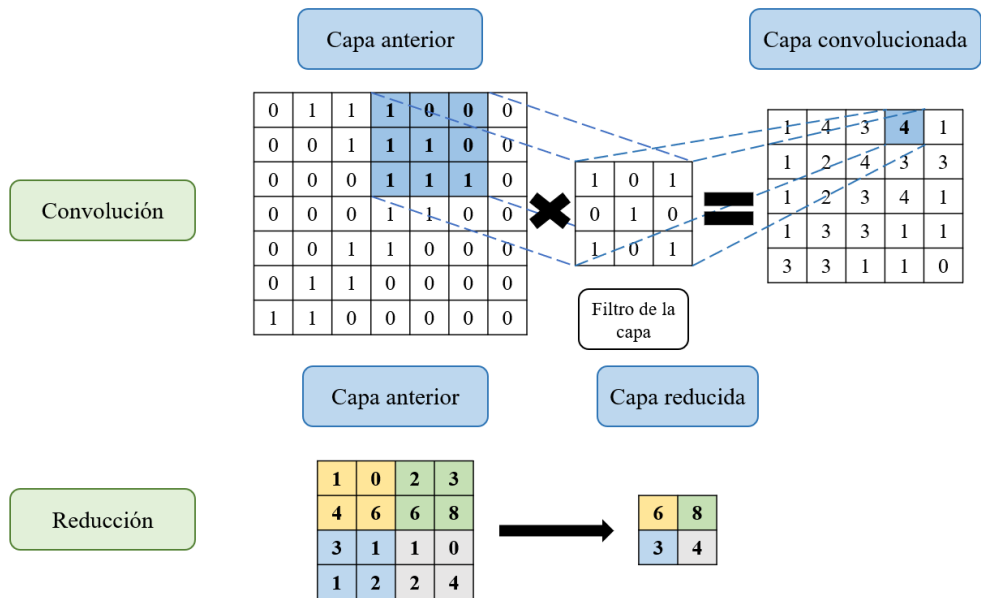


Figura 13: Etapas de reducción y convolución detalladas de una red neuronal convolucional. Adaptado de [47].

Algoritmos para la selección de variables

El objetivo de este tipo de algoritmos es escoger el mejor modelo incluyendo o excluyendo una sola variable predictora en cada paso, de acuerdo con ciertos criterios. Los algoritmos más utilizados son los siguientes:

- Métodos Forward: Se comienza con un modelo muy sencillo y se van agregando términos, es decir, se introduce la variable más significativa hasta un cierto criterio de detención.
- Métodos Backward: El punto de inicio es un modelo muy complejo, que incluye todos los efectos que pueden influir en el sistema, y se va eliminando la variable menos influyente, hasta cierto criterio de detención.
- Métodos Stepwise: Combina los dos procedimientos anteriores, comienza con una introducción progresiva de variables, pero luego se plantea si en el nuevo modelo las variables incluidas deben permanecer en él.
- Best subset: Consiste en formar todos los posibles subconjuntos de variables y efectuar todas las posibles regresiones, para luego a partir de un criterio escoger el mejor. Su gran inconveniente es el gran volumen de cálculo para explorar todas las soluciones.

Método de mínimos cuadrados penalizados: Método Lasso

Para disminuir el sobreajuste, tanto para el algoritmo jackknife como para la validación cruzada, se utiliza un parámetro de penalización, haciendo que alguna componente del vector de parámetros $\beta = (\beta_1, \dots, \beta_p)$ sea cero. Este método consiste en estimar el vector de parámetros β , minimizando la siguiente expresión:

$$\sum_{i=1}^n (y_i - x_i' \beta)^2 + \lambda \sum_{j=1}^p p_\lambda(|\beta_j|) \quad (4.3.5)$$

Donde p_λ corresponde a la función de penalización y λ es el parámetro de penalización. El vector de parámetros β que minimiza la expresión X se denota $\hat{\beta}_n$. Con esto, las tres condiciones deseables que debe cumplir el método de penalización son:

- Esparsidad: Capacidad de fijar coeficientes en cero.
- Continuidad: Sin discontinuidades en el rango de aplicación de los datos.
- Insesgadez: Evitar el sesgo, en especial para valores grandes de los coeficientes β_j .

El método Lasso (*last absolute shrinkage and selection operator*) combina la contracción de algunos parámetros hacia cero y la selección de variables, penalizando los coeficientes de regresión. Este método estima el vector de coeficientes β minimizando el siguiente problema de mínimos cuadrados penalizados:

$$\min_{\beta} \sum_{i=1}^n (y_i - x_i' \beta)^2 + \lambda \sum_{j=1}^p |\beta_j| \quad (4.3.6)$$

Cuanto mayor es el valor de λ , mayor es la penalización en los coeficientes de regresión y más son contraídos éstos hacia cero. Para hallar el valor adecuado para el parámetro de penalización λ existen diversos métodos, algunos de ellos son:

- Validación cruzada: Sea $\hat{\beta}_n^{(v)}$ el estimador de β , T^v el conjunto de datos de prueba y $T - T^v$ el conjunto de datos de entrenamiento. El estimador del parámetro λ se denota como $\hat{\lambda}$ y será el que minimiza la siguiente expresión:

$$CV(\lambda) = \sum_{v=1}^k \sum_{(y_k, x_k) \in T^v} \{y_k - x_k' \hat{\beta}_n^{(v)}(\lambda)\}^2 \quad (4.3.7)$$

- Validación cruzada generalizada: En este caso, se busca el parámetro λ que minimice la siguiente expresión:

$$GCV(\lambda) = \frac{\|Y - X\hat{\beta}(\lambda)\|^2}{n \left(1 - \frac{df(\lambda)}{n}\right)^2} \quad (4.3.8)$$

Donde $df(\lambda) = tr(B(\lambda))$, siendo tal que $\hat{Y} = B(\lambda)Y$.

3.9 Validación de resultados obtenidos de algoritmos de Machine Learning

Para mejorar muchos de los atributos antes descritos, los cuales son aspectos claves para el diseño de un buen modelo, se puede utilizar métodos de remuestreo, los cuales son métodos para la estimación de parámetros que no suponen una distribución teórica a priori [48]. A partir de las series de datos obtenidas por remuestreo se construye una distribución empírica de los parámetros que se desea estimar, lo que permite una evaluación de la estabilidad de los valores encontrados y su rango de aplicación.

Los 3 métodos de remuestreo más utilizados son: Bootstrap, validación cruzada y Jackknife [49]. El método de Bootstrap consiste en aproximar la función de distribución desconocida, F , de los datos observados mediante muestreo aleatorio con reemplazo obteniendo \hat{F} . El procedimiento determina un conjunto de k muestras del mismo tamaño sobre las cuales se obtiene el valor estadístico a inferir. Si el proceso es repetido se puede obtener una distribución simulada del valor estadístico [50]. Sin embargo, para el estudio de la calidad de las redes neuronales se utiliza un algoritmo de validación cruzada; mientras que para el algoritmo de programación genética se usa el método de Jackknife, junto a un ajuste de parámetros.

Validación de redes neuronales artificiales

Para las redes neuronales artificiales se implementa el método de validación cruzada, el cual sigue el siguiente procedimiento: Se divide aleatoriamente

el conjunto de datos en dos partes iguales, la primera mitad es utilizada para ajustar los parámetros, y luego se usa este modelo con sus respectivos parámetros para predecir los datos de la segunda mitad [49]. En general, la predicción de los segundos datos no es tan buena en comparación a usar todos los datos observados, por lo que en ese caso la capacidad de predicción observada era optimista. La extensión de este procedimiento consiste en dividir los datos experimentales en k grupos, usar $k - 1$ grupos para entrenar el modelo y el grupo restante utilizarlo para medir el error de predicción. Así, se puede repetir este proceso k veces con distintos grupos de validación y el error de predicción total como el promedio de los errores obtenidos en los k subconjuntos. A esto se le conoce como k-fold cross-validation [51].

Este método se puede ajustar dentro de la estructura de la red neuronal, a través de los parámetros *performFcn* y *performParam.regularization*, las cuales son funciones predefinidas de Matlab para la manipulación de parámetros en redes neuronales (se incluyen en la Toolbox: *Neural Network Designer* de Matlab). En el caso de *performFcn*, ésta corresponde a la función que regula el ajuste con los datos experimentales, penalizando fuertemente a los modelos inexactos. El valor de *performParam.regularization* es un número entre 0 y 1, y representa la proporción de rendimiento atribuido a los pesos de las neuronas con respecto al sesgo del modelo. Un valor alto para este parámetro implica una penalización para los grandes pesos, lo que disminuye el sobreajuste y mejora la completitud del modelo.

Validación de algoritmo de programación genética

En el caso del algoritmo de programación genética, se utiliza el método de Jackknife, el cual consiste en repetir el ajuste de parámetros una cantidad de veces igual al número de muestras N , donde se retira uno de los datos de la serie en cada iteración y utilizando los nuevos parámetros encontrados se predice el valor del dato retirado, análogo a una validación cruzada con N subconjuntos. De esta manera en cada iteración se calcula el error de predicción del modelo, permitiendo que al final del proceso se tenga una estimación del error de generalización a través del valor Jackknife Mean Square Error (MSE_{jk}), presentado en la ecuación 4.3.2, donde Y_i

corresponde al valor experimental retirado en la i -ésima iteración, y \hat{Y}_i^{-i} es el valor predicho para esta muestra [52]. Así, se logra comparar la capacidad de generalización de diferentes modelos.

$$MSE_{jk} = \sum_{i=1}^N abs(\hat{Y}_i^{-i} - Y_i) \quad (4.3.2)$$

Con el método de Jackknife también se puede calcular los pseudo-valores mediante la ecuación 4.3.3, donde a corresponde al valor original de interés, N es la cantidad de datos en la serie y a_i es el parámetro obtenido en el ajuste al retirar Y_i . Esto corresponde a un estimador no sesgado de los parámetros obtenidos en la etapa de ajuste, por lo que es posible utilizar estos pseudo-valores para determinar los parámetros de interés.

$$a^* = \frac{1}{N} \sum_{i=1}^N (N a - (N - 1) a_i) \quad (4.3.3)$$

Por otro lado, la desviación estándar de los pseudo-valores se utiliza comúnmente para estimar los intervalos de confianza de los parámetros analizados [52]. Estos intervalos se calculan con la ecuación 4.3.4, donde $t_{\alpha, \nu}$ es el valor t de Student para un valor crítico $\alpha = 0,05$, ν grados de libertad y $\hat{\sigma}_{a^*}$ corresponde al error estándar del pseudo-valor [52].

$$CI(a^*) = a^* \pm t_{\alpha, \nu} \hat{\sigma}_{a^*} \quad (4.3.4)$$

Este método aproxima de mejor manera los valores estadísticos deseados en comparación con el método de validación cruzada, ya que no sacrifica una gran parte de los datos de entrenamiento. No obstante, es más costoso computacionalmente debido a la cantidad de veces que es necesario ajustar el modelo [50]. Es por esto que presenta un buen rendimiento en casos donde no se tiene una gran cantidad de datos observados.

4. Metodología

En la Figura 14 se puede observar el esquema de flujo de la metodología de trabajo para el desarrollo de la memoria.

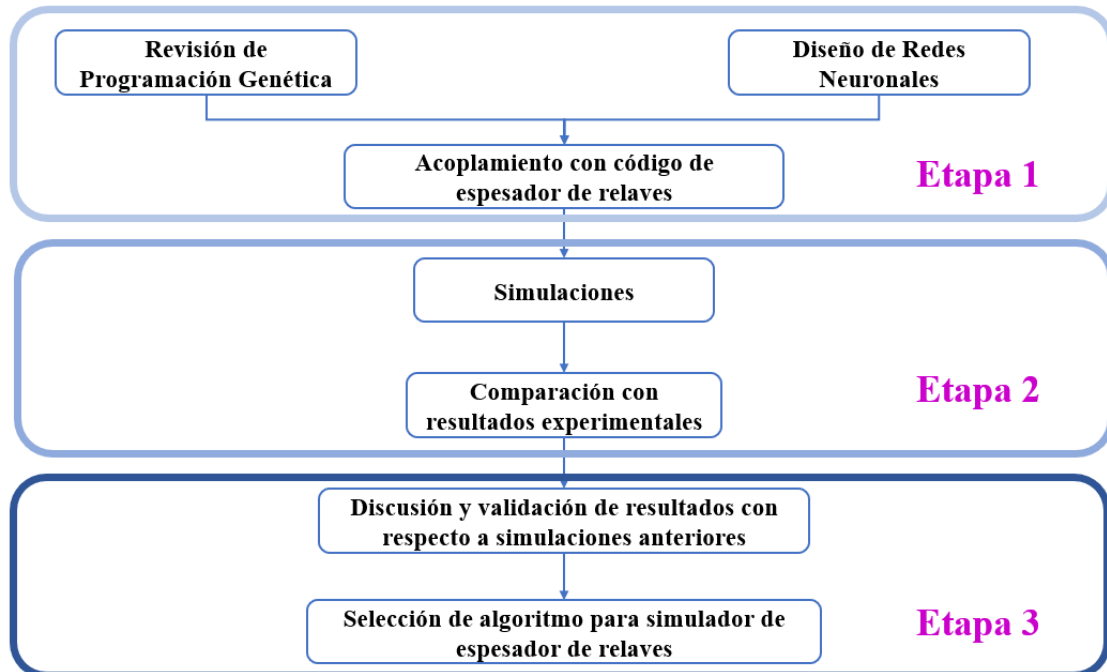


Figura 14: Diagrama de metodología utilizada.

- Etapa 1: En primer lugar, se debe hacer una revisión exploratoria de los datos, analizando si éstos podrán entregar información útil para el estudio a partir de su distribución, lo cual es necesario para utilizar los métodos de remuestreo en etapas posteriores. Además, se deberá escoger los parámetros para el diseño de los algoritmos de Machine Learning. Por un lado, para el algoritmo de programación genética se debe definir parámetros como la población inicial, método de selección de padres, índice de mutación, etc., mientras que para las redes neuronales se debe elegir el tipo de estructura, el número de neuronas de cada capa, los métodos de entrenamiento, entre otros.
- Etapa 2: La siguiente etapa corresponde a acoplar cada uno de los modelos ya diseñados con el código de simulación de espesadores de relaves, por lo que se deberá conectar adecuadamente los algoritmos.

Se realizan simulaciones para todas las muestras de relaves, registrando el tiempo de cálculo, recursos utilizados, rango de aplicación, entre otros. Con esta información, junto a parámetros estadísticos asociados (coeficiente de correlación R^2 , error cuadrático medio, error porcentual, etc.) se comparan el algoritmo de programación genética, la red neuronal convolucional y la red neuronal recurrente.

- Etapa 3: Finalmente, estos resultados obtenidos junto a la comparación de los modelos servirán para seleccionar al algoritmo que tenga un mayor rendimiento para este problema. Con esto, se espera obtener un simulador de espesadores de relaves más certero y con un mayor rendimiento a nivel informático y operacional, con respecto a las simulaciones reportadas por la empresa, las cuales están a disposición de este trabajo, pudiendo constatar los resultados teóricos con los resultados experimentales disponibles.

4.1 Definición de variables de entrada y salida de algoritmos de Machine Learning

Para realizar un apropiado análisis previo de la dispersión de las series de datos experimentales se debe observar 2 herramientas: los diagramas de cajas y bigotes (*box-and-whisker plots*) y los histogramas. Ambos son complementarios en cuanto a la información entregadas, siendo uno el que muestra los principales datos estadísticos de una muestra de datos y el otro una visualización de la distribución de dicha muestra.

Para el entrenamiento de los algoritmos de Machine Learning se utilizan los datos experimentales de 15 muestras relaves distintas. Tanto para las redes neuronales como para el algoritmo de programación genética se utiliza el mismo set de variables de entrada y salida. Para el caso de las variables de entrada, se escoge como variables que permiten caracterizar el proceso de sedimentación a los porcentajes de las 4 arcillas más relevantes por su efecto y presencia en las muestras, y el valor de D_2 de la muestra, que corresponde al porcentaje de sólidos que es capturado al pasar una muestra

por un tamiz de 2 [micras]. Así, el vector de entrada para todos los casos es el siguiente:

- x_1 : % de caolinita
- x_2 : % de clorita
- x_3 : % de esmectita
- x_4 : % de moscovita
- x_5 : D_2

A continuación, se entregan los valores de estas variables para las muestras utilizadas:

Tabla 4: Valores de variables de entrada de las muestras utilizadas.

Muestra	x_1	x_2	x_3	x_4	x_5
R01	13,90	8,90	48,20	29,00	1,84
R02	61,40	0	1,70	36,80	1,63
R03	32,80	3,90	28,90	34,40	2,10
R04	28,30	4,10	33,90	33,70	2,20
R05	27,40	10,90	38,90	22,80	2,02
R06	21,40	9,70	44,90	24,00	2,07
R07	65,20	6,30	2,20	26,30	1,77
R08	94,20	0	0	5,80	1,17
R09	61,10	18,80	0,30	19,70	1,56
R10	66,60	12,70	1,80	18,90	1,61
R11	14,70	32,80	36,40	16,10	0,74
R12	31,90	17,20	26,40	24,50	0,72
R13	30,55	4,0	31,40	34,05	2,15
R14	24,40	10,30	41,90	23,40	2,05
R15	73,50	8,36	0,83	17,26	1,40

Por otro lado, las variables de salida corresponden a los parámetros de interés de las funciones constitutivas que se desea obtener. Así, al igual que antes, todos los algoritmos tienen también el mismo set de variables de salida. Sin embargo, para evitar correlaciones erróneas, el entrenamiento de cada red neuronal y expresión obtenida a partir de programación genética se

hace de independiente. En la Figura 15 se muestra un esquema básico del entrenamiento de los algoritmos de Machine Learning, haciendo énfasis en la independencia en la obtención de las variables de salida.

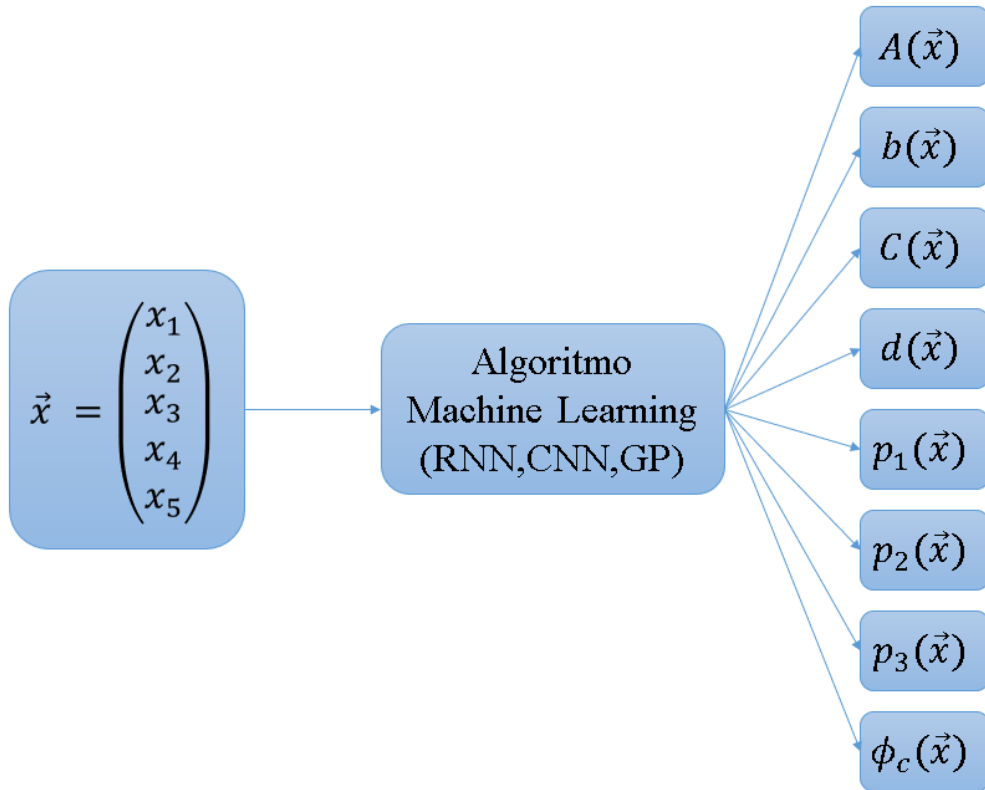


Figura 15: Diagrama de flujo de variables en algoritmos de Machine Learning.

4.2 Entrenamiento de redes neuronales

En la Tabla 5 se presentan los parámetros y criterios principales utilizados para el diseño del algoritmo, las cuales fueron escogidos en base a información bibliográfica y características propias del problema.

Tabla 5: Parámetros utilizados en el algoritmo de red neuronal.

Parámetros	Valor o criterio
Número de capas	2
Número de neuronas capa 1	10

Número de neuronas capa 2	1
Función de transferencia capa 1	Logsig
Función de transferencia capa 2	Tansig
Función de performance	Cross-entropy
Nivel de regularización de performance	0,7
Rango de normalización	[-1,1]
Porcentaje de entrenamiento	60%
Porcentaje de validación	20%
Porcentaje de prueba	20%
Función de entrenamiento	Levenberg-Marquardt

Para este trabajo se ha usado el algoritmo de Levenberg-Marquardt para entrenar las redes neuronales. Este algoritmo es una modificación del método de Newton, diseñado para acercarse en segundo orden, entrena a una gran velocidad sin tener que calcular la matriz Hessiana. Requiere mayor esfuerzo computacional que otros algoritmos disponibles en el código de Matlab, pero es el más eficiente para el entrenamiento de redes neuronales pequeñas [53]. En las Figuras 16 y 17 se observan los esquemas generales de las redes neuronales recurrentes y convolucionales diseñadas, respectivamente. Se puede observar que el número de variables de entrada a la red neuronal, en la primera capa la función de activación sigmoidea logarítmica con 10 neuronas, la segunda capa con 1 neurona y una función de activación tangente sigmoidea, y finalmente 1 variable de salida. Para el caso de la red neuronal recurrente se puede apreciar en la Figura 16 cómo la salida de la segunda capa es una de las entradas de la capa 1, creando un ciclo de retroalimentación, mientras que en la Figura 17 se puede observar el filtro de capas de convolución y reducción como paso intermedio entre la capa 1 y 2.

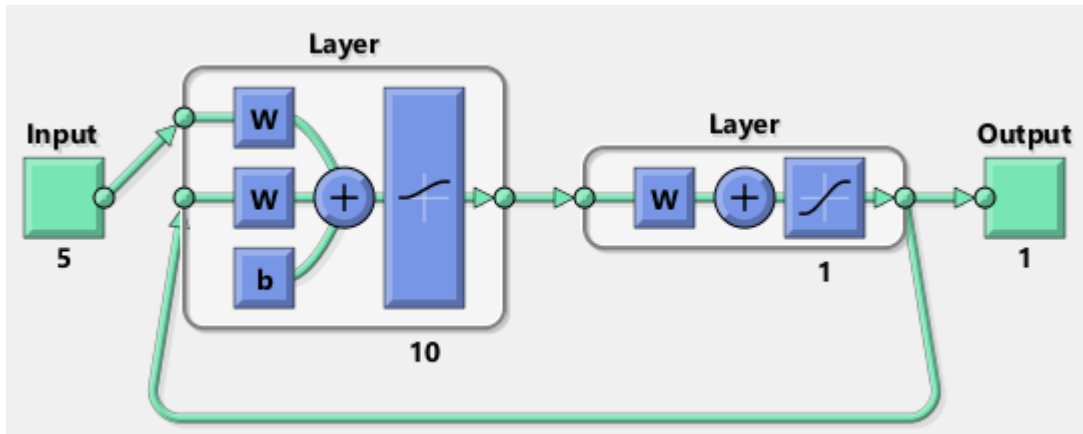


Figura 16: Diagrama de la red neuronal recurrente diseñada.

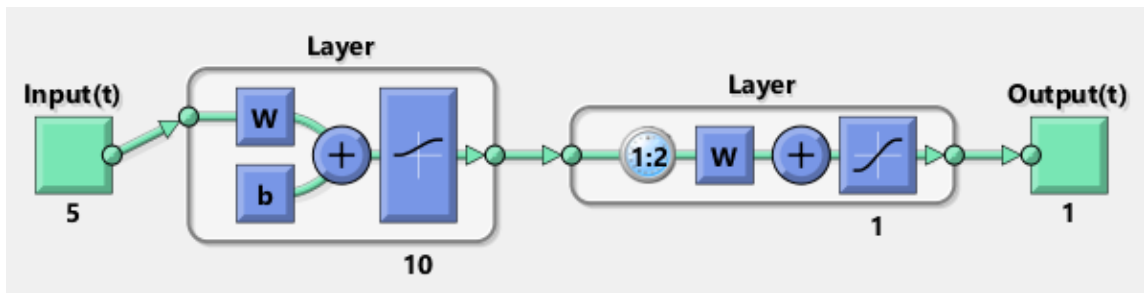


Figura 17: Diagrama de red neuronal convolucional diseñada.

4.3 Implementación del algoritmo de programación genética

La forma utilizada para la programación genética es la versión clásica del algoritmo, es decir, donde la función de fitness a optimizar es la minimización del Mean Square Error (MSE), definido en la ecuación 4.3.1.

$$MSE = \sum_{i=1}^n abs(\hat{Y}_i - Y_i) \quad (4.3.1)$$

En la Tabla 6 se presentan los parámetros y criterios principales utilizados para el diseño del algoritmo, las cuales fueron escogidos en base a información bibliográfica y características propias del problema.

Tabla 6: Parámetros utilizados en el algoritmo de programación genética

Parámetros	Valor o criterio
Tamaño de la población	500
Número de generaciones	50
Probabilidad de recombinación	0.9
Probabilidad de mutación	0.1
Elitismo o reproducción	Mantener sólo el mejor
Conjunto de funciones	$\sqrt{\quad}, \ln, +, -, \cdot, /, e^a, a^b$
Conjunto de terminales	Variables: x_1, x_2, x_3, x_4, x_5 ; Constantes: 1,5,10
Población inicial	Ramped-Half-and-Half
Límite de profundidad del árbol	17
Métodos de selección	Lexicographic parsimony pressure

Se realizaron 20 repeticiones del algoritmo de Programación, donde cada una de estas repeticiones contenía un modelo capaz de representar los datos experimentales. Para escoger los modelos más representativos de estos datos se utilizaron distintos criterios. En primer lugar, se analiza la función de fitness de los modelos obtenidos, se mide su capacidad de representar los datos del conjunto de validación, y se estudia la capacidad de generalización de los modelos, para así los parámetros utilizando el método de Jackknife para el caso de la programación genética, y cross-validation para el diseño de las redes neuronales artificiales.

4.4 Validación de resultados

Una vez construido un modelo capaz de representar una serie de datos experimentales es necesario evaluar su calidad. La capacidad de generalización y la calidad de un modelo se pueden evaluar a partir del estudio de los siguientes atributos:

- Expresividad: Grado en que una técnica de modelado es capaz de representar un gran número y tipos de dominios de aplicación.
- Arbitrariedad: Grado de libertad de las variables en el modelo para un mismo dominio.
- Idoneidad: Grado en el que una técnica de modelado se adapta a un tipo específico de dominio.
- Comprensibilidad: Facilidad para comprender el funcionamiento interno del modelo diseñado.
- Completitud: Grado en el que todas las variables participan en la construcción del modelo.
- Eficiencia: Grado en que el modelo utiliza recursos (de cualquier tipo) para su ejecución y diseño. Para este caso, se refiere a recursos computacionales de memoria y tiempo.
- Efectividad: Grado en el que el proceso de modelado logra representar el fenómeno de estudio.

5. Resultados y discusión

5.1 Dispersión y Variabilidad de inputs y targets

Para la construcción de los algoritmos de Machine Learning se utilizaron variables que caracterizan a cada muestra de relave. De esta forma, y a partir de la experiencia experta de los ingenieros de la planta, se escogió trabajar con el porcentaje de los 4 minerales más abundantes, 2 parámetros relacionados con las propiedades granulométricas y también con la densidad específica del sólido. Sin embargo, en una primera instancia, se realiza un estudio de la dispersión de los datos numéricos de dichas variables. En la Figura 18 se aprecia un diagrama de caja, donde se representan parámetros estadísticos de una serie de datos mediante cuartiles.

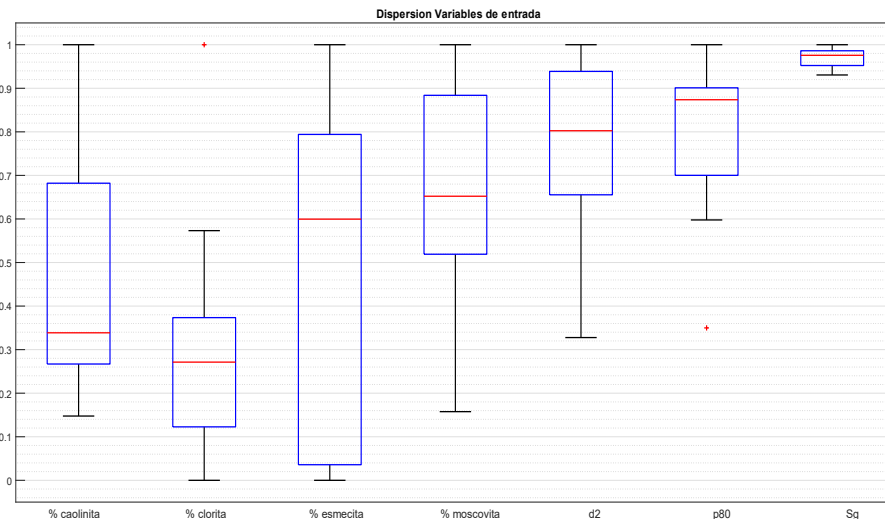


Figura 18: Dispersión de las variables de entrada para los algoritmos de Machine Learning

En la Figura 18 se observan los cuartiles de las variables de entrada a los algoritmos de Machine Learning. Este tipo de diagrama representa la dispersión de estas variables, y en este caso todas las variables se representan en el mismo eje debido al proceso de normalización de los

datos de entrada, el cual es un paso necesario para el uso de estos algoritmos. Con respecto al porcentaje de caolinita, esmectita y moscovita, junto al valor de d_2 y la gravedad específica (Sg), se observa que estas variables no presentan datos anómalos, por lo que tienen una distribución típica y una distribución de cuartiles balanceada. Sin embargo, para el caso de Sg , la distribución de cuartiles es muy acotada, lo que a nivel matemático representa que los datos numéricos no reportan una diferencia significativa, por lo que se decide realizar un test de hipótesis, validando que los datos no son significativamente distintos. A nivel de procesos esto significa que la gravedad específica es prácticamente igual para las muestras de relaves, por lo que no corresponde una variable que permita caracterizar y diferenciar el comportamiento de las muestras. Debido a esto, se decide descartar dicha variable. Por otro lado, $p80$ y el porcentaje de clorita presentan datos anómalos en sus respectivas series de datos; no obstante, la distribución de cuartiles para el caso de $p80$ es notoriamente irregular, a nivel matemático existe un sesgo hacia el extremo inferior del rango de datos, lo que se traduce que varios de los datos de la serie presentan un valor numérico similar. A nivel de procesos esto implica que un gran número de muestras tienen propiedades similares en cuanto a su granulometría, mientras que otros se alejan de éstas, y al no estar distribuidos uniformemente estos datos, se distorsiona la información que permitiría captar el modelo. Debido a esto, se decide eliminar esta variable dentro de las entradas de los algoritmos de Machine Learning. Con todo lo anterior, las variables de entradas para el diseño de los algoritmos de Machine Learning son:

x_1 :	Porcentaje de caolinita
x_2 :	Porcentaje de clorita
x_3 :	Porcentaje de esmectita
x_4 :	Porcentaje de moscovita
x_5 :	Porcentaje de sólido captado en un tamiz de 2 micras D_2

Por otro lado, se puede realizar el mismo ejercicio estudiando la información estadística de las variables de salida para el entrenamiento de los algoritmos (target), presentado en la Figura 19.

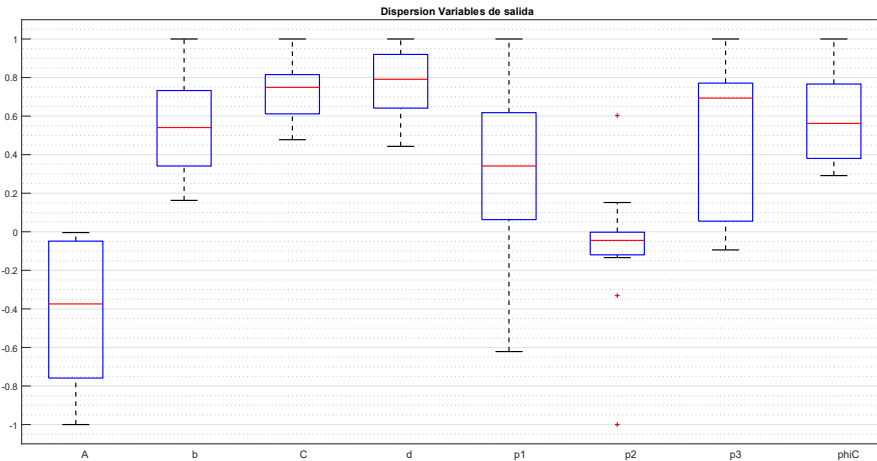


Figura 19: *Dispersion del target para los algoritmos de Machine Learning*

Con esto, se pueden hacer 4 observaciones:

- Para A, b, C, d y ϕ_c , se presenta series sin datos anómalos y con una distribución equitativa de sus cuartiles. Esto significa a nivel matemático que estas variables no presentan problemas a priori para ser modeladas, mientras que a nivel de procesos la medición y elección de puntos de medición se hizo de manera correcta, lo que mejora la capacidad del modelo para captar la información deseada.
- Para p_1 se presenta un amplio rango donde los datos numéricos son típicos. Esto significa a nivel matemático que los datos están dispersos de manera equitativa en un gran rango de aplicación, que a nivel de procesos se traduce en que las muestras tienen un comportamiento muy distinto entre ellas con respecto a la dosificación de floculante, es decir, no se observa un comportamiento notorio y definido para este coeficiente, lo que implica para el análisis una leve dificultad para modelar dicha variable.
- Para p_2 se observa la presencia de 3 puntos anómalos de la distribución típica de los datos de la serie, lo que presentará problemas para los métodos de remuestreo utilizados puesto que no presentan buen rendimiento para series no típicas. A nivel de procesos, estos puntos anómalos corresponden a muestras con valores

que se alejan de la tendencia general de los demás datos, por lo que interfieren en el diseño del modelo.

- Por último, la variable p_3 muestra una distribución asimétrica en cuanto a sus cuartiles, lo que a nivel de procesos significa que los datos medidos pueden tener un sesgo hacia el extremo superior de los datos, e implica una dificultad para modelar la variable, aunque leve como en el caso de p_1 .

Estos puntos se deben tener en cuenta para el análisis y discusión de los resultados arrojados por los algoritmos de Machine Learning, ya que las discrepancias que éstos puedan presentar se deben, en parte, a lo observado en esta etapa y no al diseño del algoritmo en sí.

5.2 Implementación de algoritmo de programación genética

Consideraciones

En esta sección se analizarán cada una de las expresiones obtenidas para las variables a partir de programación genética. Sin embargo, es necesario hacer algunas consideraciones sobre las restricciones matemáticas del modelo.

- Las operaciones básicas utilizadas para el diseño de estos modelos fueron: suma, resta, multiplicación, división, logaritmo natural, raíz cuadrada, exponencial y potencia. Se descartaron otras funciones disponibles, como por ejemplo funciones trigonométricas, y periódicas, ya que no corresponde a un comportamiento esperado con la teoría.
- La operación de división posee una condición contra indeterminación, para que los valores obtenidos pertenezcan a números naturales. En este caso:

$$\text{división}(a, b) = \begin{cases} a & \text{si } b = 0 \\ a/b & \text{si } b \neq 0 \end{cases} \quad (5.2.1)$$

- La operación de logaritmo natural también posee una condición contra indeterminación, para que los valores obtenidos pertenezcan a números naturales. En este caso:

$$\text{logn}(a) = \begin{cases} 0 & \text{si } a = 0 \\ \ln(|a|) & \text{si } a \neq 0 \end{cases} \quad (5.2.2)$$

- La operación de raíz cuadrada también posee una condición contra indeterminación, para que los valores obtenidos pertenezcan a números naturales. En este caso:

$$\text{sqrt}(a) = \begin{cases} 0 & \text{si } a \leq 0 \\ \sqrt{a} & \text{si } a > 0 \end{cases} \quad (5.2.3)$$

Expresiones obtenidas

A continuación, se presentan las expresiones matemáticas que representan los modelos obtenidos a partir del algoritmo de programación genética.

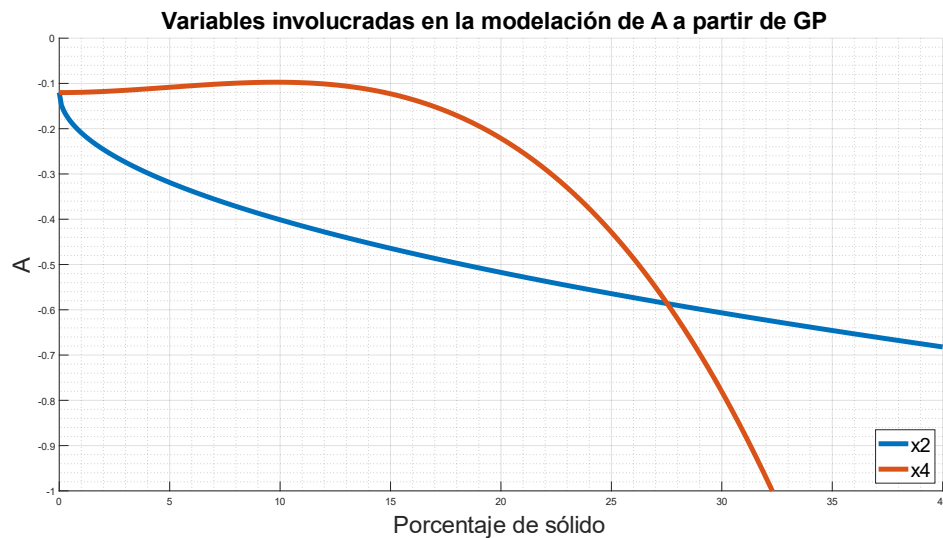


Figura 20: Variables involucradas en la modelación de A a partir de GP.

$$A(x) = -\sqrt{x_2}x_4(1,5 \cdot 10^{-5}x_4^2) - 14,8 \quad (5.2.4)$$

Para la ecuación (5.2.4) se observa que el modelo propone que el coeficiente A depende de sólo 2 variables: el porcentaje de clorita, en la cual tiene una forma de raíz cuadrática con respecto a esta variable, y el porcentaje de moscovita, con un orden cúbico de crecimiento. Este comportamiento distinto entre ambas variables se explica a partir de la relevancia que éstas tienen sobre A . En la Figura 20 se observa que la tasa de decrecimiento es mayor para x_4 , siendo casi el triple en comparación a x_2 , por lo que principalmente el comportamiento está dominado por el porcentaje de moscovita. Este modelo no logra representar las variaciones del porcentaje de caolinita, el porcentaje de esmectita y el valor de D_2 , y sin embargo, logra tener un coeficiente de correlación de 0,8979 con el uso de 2 variables de entrada. La expresión se condice con el comportamiento esperado puesto que a medida que aumenta alguno de los porcentajes involucrados, el valor del coeficiente A disminuye.

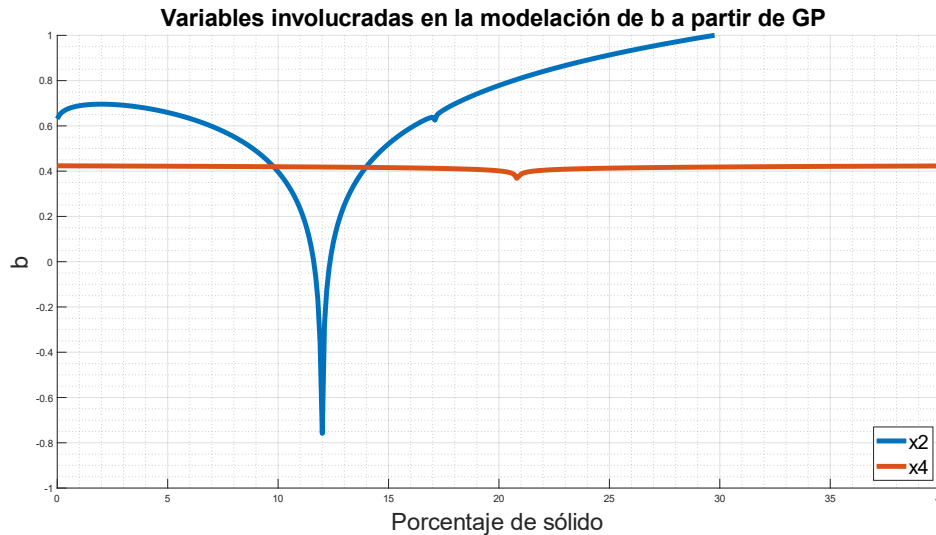


Figura 21: Variables involucradas en la modelación de b a partir de GP.

$$b(x) = 0,24 \ln \left(-11,9 \left(\ln \left(\ln \left((228,9 - 19,8x_4) \ln(-6,9(4,4 + 2x_2 - \ln(2x_4))) \right) + x_2 \right) + x_2 \right) \right) + x_2 \quad (5.2.5)$$

Con respecto a la ecuación (5.2.5), se puede notar que las variables de entradas que participan en la expresión son: el porcentaje de clorita y el porcentaje de moscovita. En la Figura 21 se observa que, en cuanto a relevancia, x_2 afecta en mayor medida el comportamiento de b , En este caso el análisis es más complejo ya que las variables se presentan de forma logarítmica, aunque en los gráficos sobre las tendencias respecto a las variables separadas (ver Sección 5.5) se observa un pronunciado mínimo en la mitad del rango de aplicación, para luego volver a subir, comportamiento que se puede observar para ambas variables. Para este caso el coeficiente de correlación es 0,8303. Cabe destacar que a pesar de que la expresión posea términos con logaritmo natural, al ser una función protegida de la manera en que se describe en la subsección anterior está bien definida en su dominio.

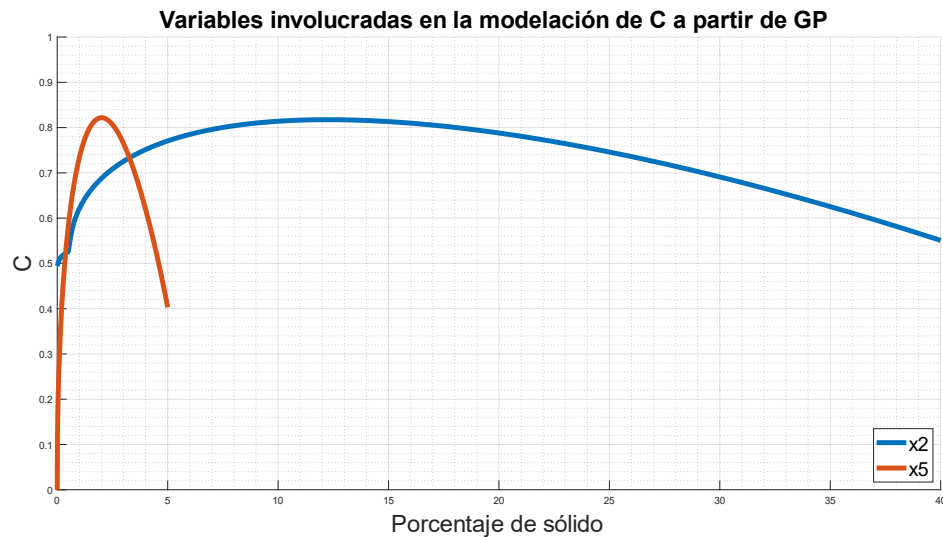


Figura 22: Variables involucradas en la modelación de C a partir de GP.

$$C(x) = -0.02 \left(x_2 - \ln(10.9x_5)(V) - 0.018 \left(\sqrt{U} - \ln \left(\sqrt{Z} \ln(x_5 \sqrt{x_5}) \sqrt{(\ln(10x_5)(V) - (\ln(U) + Z))34.6} + 1476.1 \right) \right) \right) \quad (5.2.6)$$

$$\text{con } V = \sqrt{7.9x_2}, \quad U = \sqrt{7.6 \cdot 10^{-8}}, \quad Z = x_2 - 8.9$$

En la ecuación (5.2.6) se observa que la expresión depende de las siguientes variables de entrada: porcentaje de clorita, quien para valores pequeños predomina el comportamiento logarítmico, para luego ser más

preponderante el componente lineal negativo, y también depende del valor de D_2 donde se presenta mayormente un comportamiento logarítmico con respecto a esta variable. En la Figura 22 se observa que el mayor impacto lo produce x_2 , es decir, el porcentaje de clorita, mientras que x_5 afecta en mayor medida a C para porcentajes bajos. Para este caso, el coeficiente de correlación corresponde a 0,8451, logrando capturar satisfactoriamente la información de los datos experimentales.

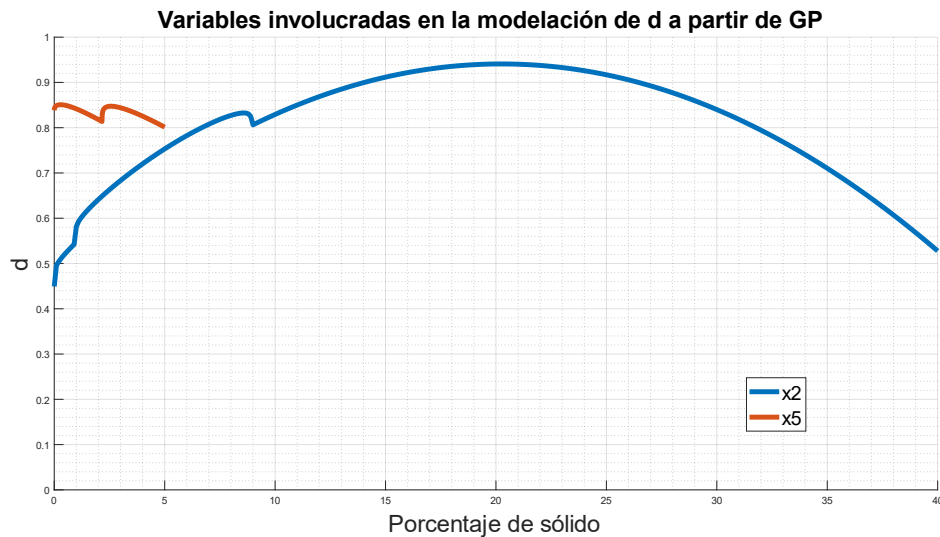


Figura 23: Variables involucradas en la modelación de d a partir de GP.

$$d(x) = \ln \left(\sqrt{2 \ln \left(17,5 \ln \left(\ln \left(x_2 + (x_2 + 36,8)(-0,02\sqrt{x_5}) \right) \right) + \ln \left(\sqrt{\ln(10 + x_2)} \right) + (x_5 x_2) \right)} \right) \quad (5.2.7)$$

La ecuación (5.2.7) depende del porcentaje de x_2 , donde se muestra un comportamiento aparentemente parabólico cóncavo, y de x_5 , donde es mayormente constante en el rango de aplicación, como se puede observar en la Figura 23. El coeficiente de correlación para esta expresión es de 0,8594. Al igual que para el caso anterior, la función está protegida por indeterminaciones en su dominio. El comportamiento está principalmente dominado por el porcentaje de clorita, por lo que una muestra con un mayor porcentaje de esta arcilla implicará una pendiente mayor en la curva de la función estrés efectivo de sólido, por lo que se producirá una mayor consolidación de sólidos en el espesador, teniendo

como punto máximo para d aproximadamente a un porcentaje de 20% de clorita, luego el valor de d disminuye para porcentajes mayores.

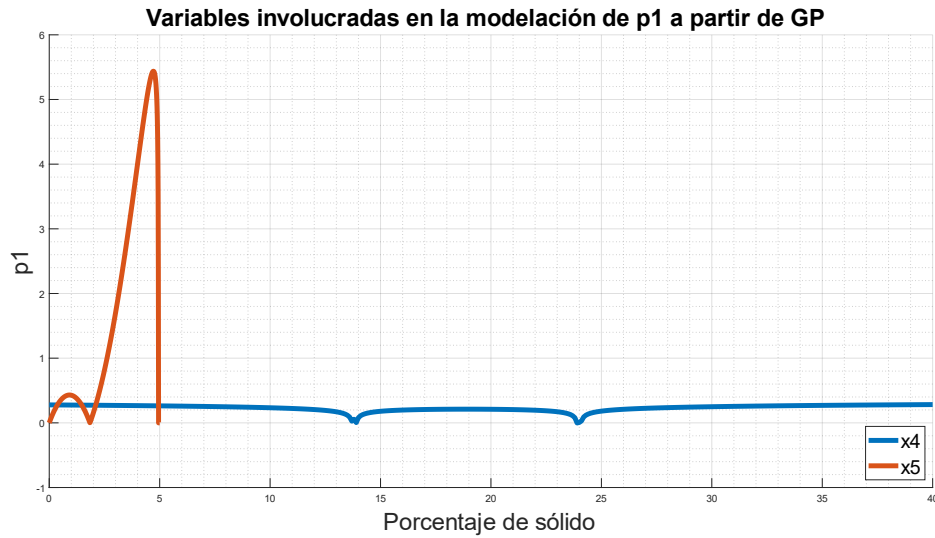


Figura 24: Variables involucradas en la modelación de p_1 a partir de GP.

$$p_1(x) = (4,8x_5 e^{-2,2(x_5 - 1,8)}) \ln \left(\ln \left(3 \left((-x_4 - 4,2) \left(\sqrt{\ln(x_5 - 5,9)} (x_4 - 13,8) \right) \right) \right) \right) \quad (5.2.8)$$

Con respecto a la ecuación (5.2.8) se puede observar que la expresión depende de las siguientes variables: el porcentaje de moscovita, que se encuentra dentro de términos logarítmicos, y el valor de D_2 , donde se tiene en primer lugar un componente logarítmico y también lineal positivo. El coeficiente p_1 simulado a partir de este modelo tiene un coeficiente de correlación de 0,7417, por lo que logra captar la información parcialmente. Para este caso, se observa en la Figura 24 que x_5 tiene una mayor relevancia sobre p_1 , siendo para x_4 prácticamente constante, por lo que para p_1 no depende del tipo de arcilla que tenga la muestra relave sino de la cantidad de material fino que ésta pueda tener.

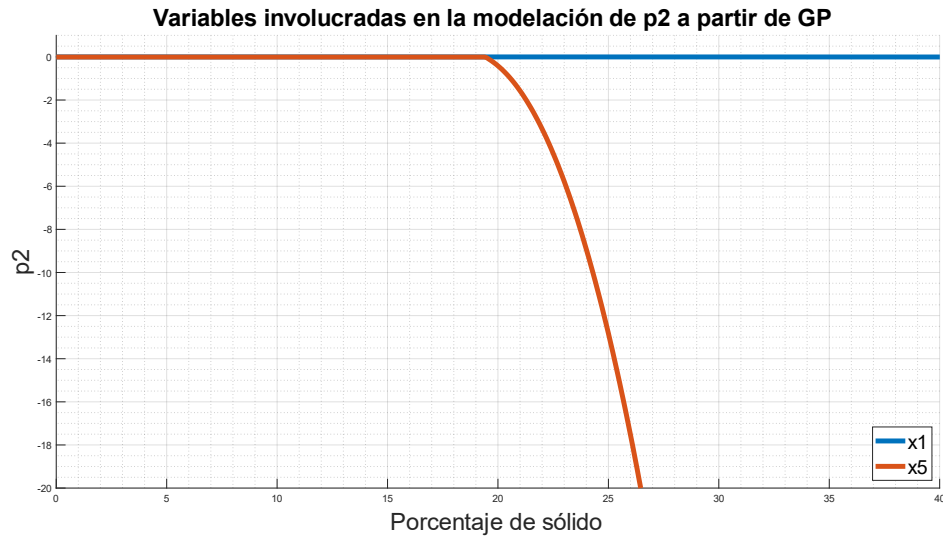


Figura 25: Variables involucradas en la modelación de p_2 a partir de GP.

$$p_2(x) = -1,4 \cdot 10^{-9} x_1^4 \left(\exp \left(-8,9 \left(\sqrt{(\sqrt{x_2} - e^{-9,4 \cdot 10^{-6}}) \sqrt{-0,003 x_1 - \ln(80,9)}} - \sqrt{\sqrt{x_2} - \ln(82,5)}} \right) \right) \right) \quad (5.2.9)$$

En la ecuación (5.2.9) se puede observar que la expresión depende de las siguientes variables: porcentaje de caolinita, donde se muestra un comportamiento polinómico de orden 4, junto a un componente exponencial, y el porcentaje de clorita, donde se observa una forma exponencial. En este caso, el coeficiente de correlación es de 0,8634, por lo que es un modelo que logra representar a los datos, aunque no captura todo el fenómeno. En la Figura 25 se observa que el porcentaje de clorita (x_2) es la variable que predomina en el comportamiento de p_2 , y esto se puede observar en la ecuación, analizando que x_1 figura como un coeficiente pre-exponencial de orden 4, mientras que x_2 aparece como un exponente en la ecuación.

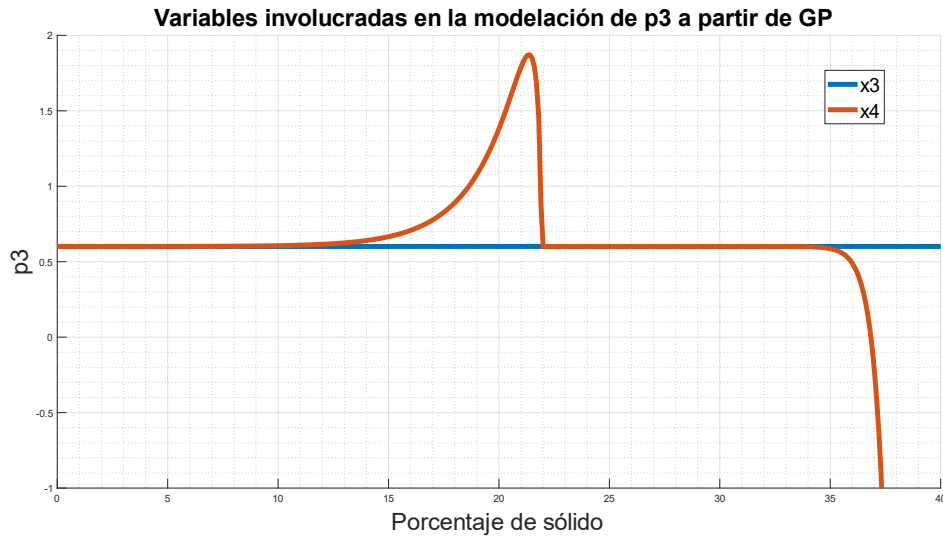


Figura 26: Variables involucradas en la modelación de p_3 a partir de GP.

$$p_3(x) = 18,9 \cdot 10^{-16} e^{2x_4} + 1,4 \cdot 10^{-6} \sqrt{e^{x_4} + 1,13} \sqrt{\left(((x_3 + 5)x_3) - (13,9 - 18,7e^{x_4})e^{x_4} \right) - e^{\sqrt{W}}} - e^W \quad (5.2.10)$$

$$\text{con } W = \sqrt{(x_3 + x_4) - 76,9}$$

Para la ecuación (5.2.10) se observa que el modelo propone que el coeficiente p_3 depende de sólo 2 variables: el porcentaje de esmectita, el cual tiene una forma mayormente de exponencial, y el porcentaje de moscovita, con un orden lineal de crecimiento. Este modelo no logra representar las variaciones del porcentaje de caolinita, el porcentaje de clorita y el valor de D_2 , teniendo un desempeño insuficiente con un coeficiente de correlación de 0,3536. En la Figura 26 se puede observar el aumento que sufre p_3 para un porcentaje de moscovita (x_4) de entre un 15% a lo 22%, mientras que para el porcentaje de esmectita (x_3) el comportamiento de p_3 es constante en todo el rango.

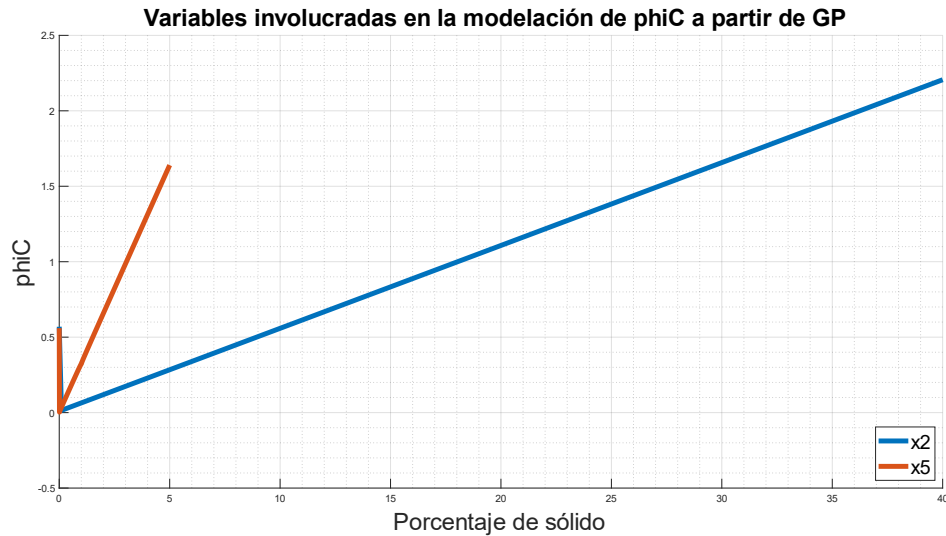


Figura 27: Variables involucradas en la modelación de ϕ_c a partir de GP.

$$\phi_c = 1,5e^{-11,5} \left(x_5 + 7,6 \left(\ln(x_5) + e^{\left(\ln(\ln(2x_5 \ln(x_5 + 9,3 \cdot 10^{-7}))) + 1,4(x_5 + e^{\ln(0,06x_2x_5) + 5,5}) \right) + 2,7} \right) \right) \quad (5.2.11)$$

Con respecto a la ecuación (5.2.11), se tiene que la expresión de la concentración crítica depende de 2 variables: porcentaje de clorita, teniendo un comportamiento principalmente lineal, y el valor de D_2 , quien también exhibe el mismo comportamiento. En la Figura 27 se observa que el efecto de x_2 es mayor, siendo una variable más relevante que x_5 para este coeficiente, aunque para valores bajos de x_5 , es decir, bajo porcentaje de material fino, éste domina en mayor medida que x_2 . Para este caso el valor del coeficiente de correlación es de 0,9694, existiendo una alta relación lineal entre los datos experimentales y los valores simulados a través de programación genética.

5.3 Implementación de redes neuronales

Redes neuronales recurrentes

En la Figura 28 se observa el efecto de cada variable de x sobre el coeficiente A . Se observa que a nivel general las variables presentan la misma relevancia sobre A , aunque entre 30% y 40% de x_3 (% de esmectita) sobresale, dominando el comportamiento de A .

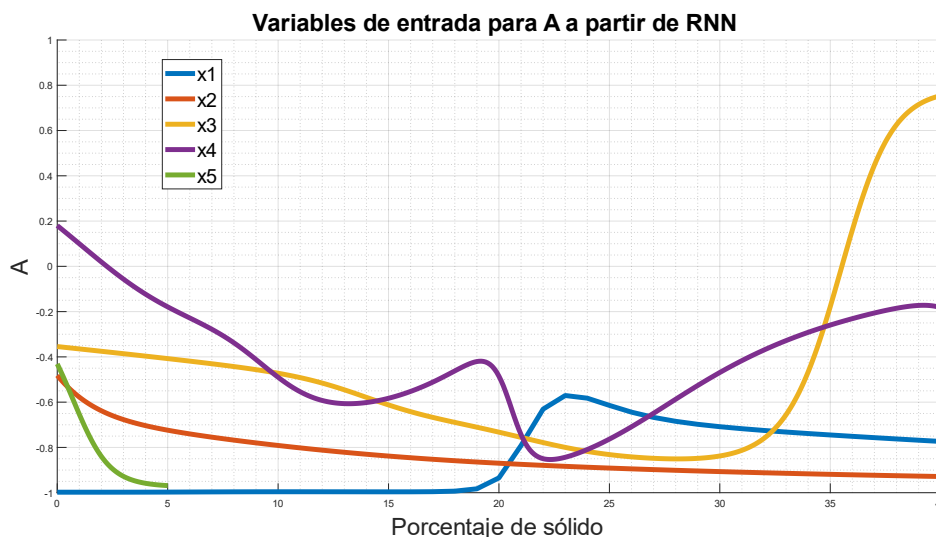


Figura 28: Variables involucradas en la modelación de A a partir de RNN.

Lo anterior se puede observar en la matriz que se presenta a continuación, donde se pueden ver los valores de los pesos correspondientes a cada neurona de cada variable de entrada x . En la columna 3, que corresponde a los pesos de las neuronas involucradas con x_3 , se observan los valores más altos en magnitud. En la matriz se puede observar que los valores tienen en general el mismo orden de magnitud. Cabe destacar que para x_5 (D_2) se tienen sólo valores negativos, lo que muestra una relación lineal inversa entre esta variable y el coeficiente A .

$$w_A = \begin{pmatrix} -0.3095 & 0.1846 & -0.3212 & 0.0717 & -0.1259 \\ 0.3675 & 0.3176 & 0.0431 & 0.0366 & -0.1217 \\ -0.4552 & 1.3041 & -0.4399 & -0.2898 & -0.1235 \\ -0.7166 & -0.5170 & -0.3710 & 0.7011 & -0.1384 \\ -0.2330 & -0.1448 & 0.1264 & 0.6092 & -0.2329 \\ -0.1312 & 0.1153 & -0.3067 & -0.3328 & -0.1618 \\ -1.0244 & -0.5215 & 0.3398 & 0.1562 & -0.1569 \\ 0.4351 & -0.6648 & -0.6830 & -1.4291 & -0.2348 \\ -1.7699 & 0.9448 & 0.1981 & 1.4550 & -0.4534 \\ -1.4502 & 2.2989 & -1.7984 & 3.6973 & -0.1056 \end{pmatrix}$$

En la Figura 29 se observa el efecto de cada variable de x sobre el coeficiente b . Se observa que a nivel general las variables presentan la misma relevancia sobre A excepto x_5 que presenta un rango en A mucho menor, en comparación a las otras variables. Por otro lado, la variable x_4 (% de moscovita) es la que presenta el rango más amplio.

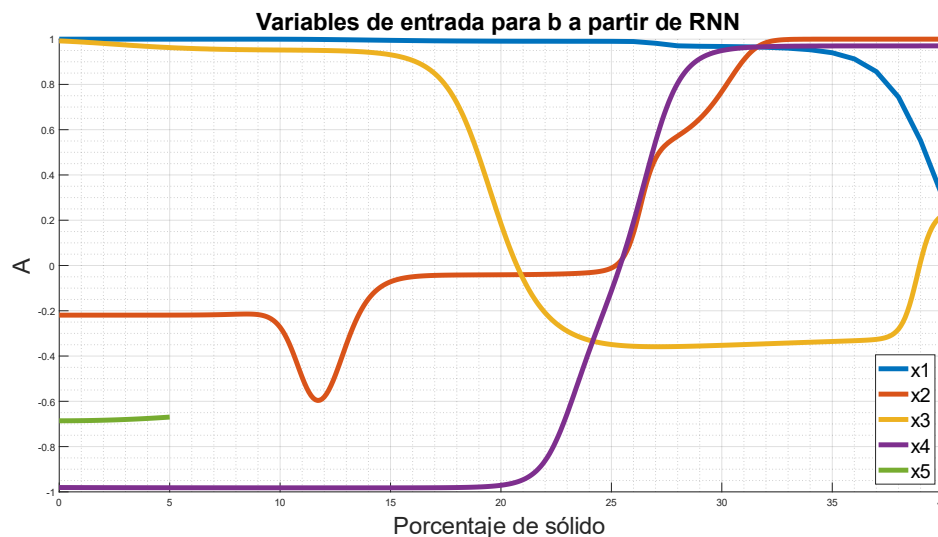


Figura 29: Variables involucradas en la modelación de b a partir de RNN.

En la matriz w_b se presentan los pesos correspondientes a cada neurona de cada variable de entrada x . Se observa en este caso que la variable x_5 presenta valores estrictamente positivos para sus pesos correspondientes, mostrando una relación directa entre x_5 y b , mientras que para x_1 y x_3 (% de caolinita y esmectita, respectivamente) se observan mayoritariamente valores negativos, reflejando una relación inversa con el coeficiente b .

$$w_b = \begin{pmatrix} -1.1048 & -0.2553 & -0.1304 & 0.4154 & 1.3640 \\ -0.6342 & 1.1641 & -0.7954 & 0.2221 & 1.3704 \\ 1.5715 & 0.8366 & -0.3247 & -0.1668 & 1.3707 \\ -0.4431 & 0.6197 & -0.2277 & 0.0367 & 1.3359 \\ -0.5649 & 1.3552 & -0.6724 & 0.8583 & 1.4469 \\ -0.1644 & -1.7394 & 0.0316 & 0.9915 & 1.0065 \\ -2.8200 & 2.7693 & 2.7988 & -0.7326 & 2.1239 \\ -0.4296 & 0.6081 & -0.2120 & 0.0085 & 1.3342 \\ -0.4141 & 0.5998 & -0.1880 & -0.0393 & 1.3396 \\ 0.0609 & 0.0031 & -0.4093 & -0.2300 & 1.3305 \end{pmatrix}$$

En la Figura 30 se observa el efecto de cada variable de x sobre el coeficiente C . Se puede ver que la variable x_1 (% de caolinita) presenta un rango mayor al de las otras variables, distanciándose de ellas en especial para valores bajos de x_1 (0-15%).

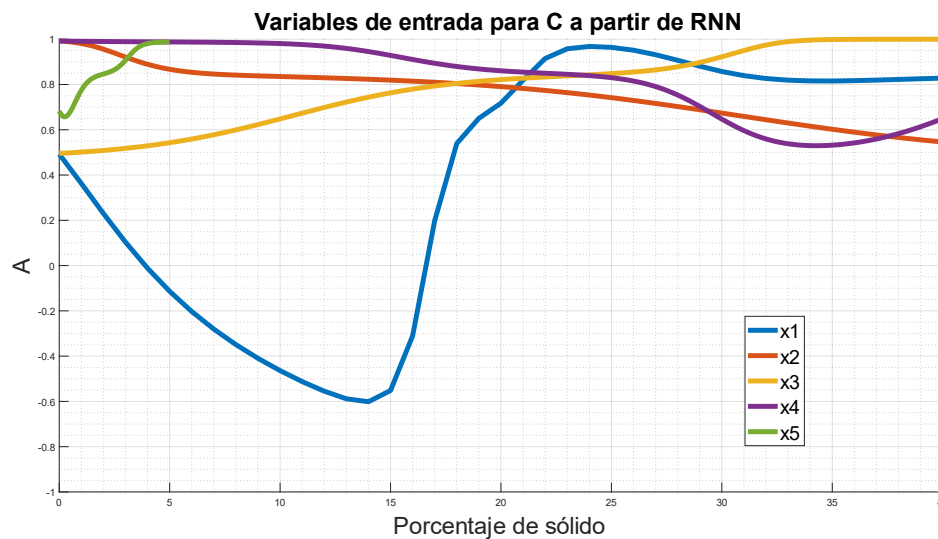


Figura 30: Variables involucradas en la modelación de C a partir de RNN.

En la matriz w_C se presentan los pesos correspondientes a cada neurona de cada variable de entrada x . Se observa en este caso que la variable x_5 presenta valores estrictamente positivos para sus pesos correspondientes, mostrando una relación directa entre x_5 y C , mientras que para x_1 y x_2 (% de caolinita y esmectita, respectivamente) se observan mayoritariamente valores negativos, reflejando una relación inversa con el coeficiente C .

$$w_c = \begin{pmatrix} 1.3488 & 0.0982 & 0.3660 & -2.2651 & 0.6258 \\ 0.2496 & -0.0344 & 0.1053 & 0.1344 & 0.5062 \\ -0.1417 & 0.6568 & 0.0911 & 0.9140 & 0.5522 \\ 0.3723 & -0.1722 & -0.3109 & 0.0494 & 0.5013 \\ -0.3206 & -0.5320 & 0.4186 & 0.3234 & 0.5681 \\ -0.2710 & -0.0851 & 0.0851 & -0.0262 & 0.5389 \\ -0.6525 & -0.3304 & -0.1344 & 0.3228 & 0.5591 \\ -0.2358 & -0.0461 & -0.0772 & -0.1019 & 0.5227 \\ -0.4917 & -0.3337 & -0.1449 & 0.3523 & 0.5581 \\ 0.6773 & 0.0824 & -0.2859 & -0.2584 & 0.5039 \end{pmatrix}$$

En la Figura 31 se observa el efecto de cada variable de x sobre el coeficiente d . Se puede ver que las variables x_2 y x_3 (% de clorita y esmectita, respectivamente) presenta un rango mayor al de las otras variables, distanciándose de ellas en especial para valores bajos en el caso de x_2 (0-10%) y valores altos para x_3 (30-40%).

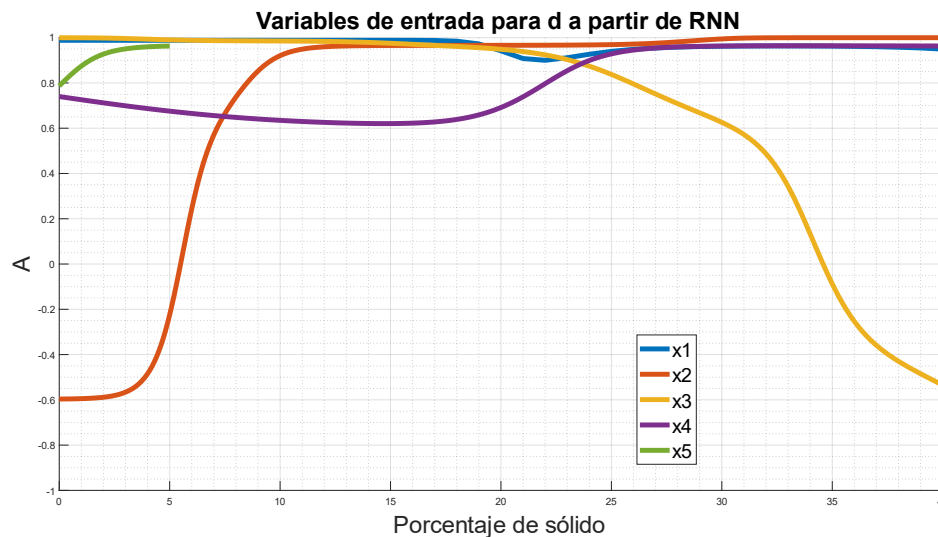


Figura 31: Variables involucradas en la modelación de d a partir de RNN.

En la matriz w_d se presentan los pesos correspondientes a cada neurona de cada variable de entrada x . Se observa en este caso que las variables x_2 y x_5 presenta valores estrictamente positivos para sus pesos correspondientes, mostrando una relación directa con d .

$$w_d = \begin{pmatrix} -0.3059 & 0.9463 & -0.3035 & 0.2698 & 0.2130 \\ 1.1495 & 0.8379 & -0.6319 & -0.7430 & 0.1981 \\ -0.1853 & 0.6940 & 0.1260 & 0.0458 & 0.2209 \\ -0.1981 & 0.7345 & 0.1317 & 0.0541 & 0.2217 \\ 0.5182 & 0.4173 & -0.8219 & -0.1020 & 0.1117 \\ -0.2080 & 1.3381 & -0.3562 & 0.2302 & 0.2590 \\ 0.0068 & 0.5146 & 0.1102 & 0.1606 & 0.2141 \\ 0.6629 & 1.3404 & -0.6091 & -0.0811 & 0.2140 \\ -0.2715 & 0.9849 & -0.4939 & 0.1313 & 0.1375 \\ -0.1713 & 1.0660 & 0.5474 & -0.3285 & 0.2247 \end{pmatrix}$$

En la Figura 32 se observa el efecto de cada variable de x sobre el coeficiente p_1 . Se puede ver que la variable x_1 (% de caolinita) presenta un rango mayor al de las otras variables, distanciándose de ellas en especial para valores entre 5 y 25%.

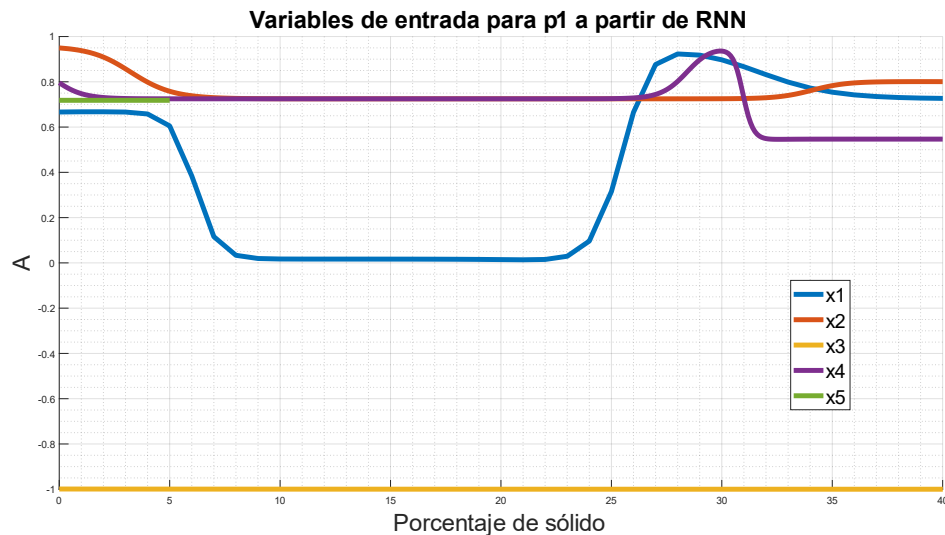


Figura 32: Variables involucradas en la modelación de p_1 a partir de RNN.

En la matriz w_{p_1} se presentan los pesos correspondientes a cada neurona de cada variable de entrada x . Se observa en este caso que las variables x_2 y x_5 presentan mayoritariamente valores negativos, mostrando una relación inversa con p_1 . La matriz muestra valores similares en magnitud entre los pesos, mostrando un balance entre las neuronas de una misma variable.

$$w_{p_1} = \begin{pmatrix} -1.3897 & -0.1884 & 1.3255 & -1.3675 & -0.8082 \\ -0.0918 & 0.2434 & -0.1550 & -0.8779 & -0.8866 \\ -0.5710 & -0.9879 & -0.5366 & 1.6492 & -0.6975 \\ 2.1363 & -0.9544 & -2.3795 & 0.2860 & -0.7454 \\ 0.1269 & -0.5768 & 0.0313 & -1.4644 & -0.8629 \\ -0.2519 & -0.3177 & -0.5812 & -0.0380 & -0.8276 \\ 1.4106 & -0.4684 & 2.9062 & -3.9109 & -0.8922 \\ 1.9025 & -0.9833 & -0.4302 & 0.3831 & -0.8187 \\ -1.1093 & 0.2490 & 0.4634 & -1.0145 & -0.8312 \\ 0.5664 & -1.2831 & 0.5700 & 0.3457 & -1.1032 \end{pmatrix}$$

En la Figura 33 se observa el efecto de cada variable de x sobre el coeficiente p_2 . Se puede ver que las variables presentan comportamientos distintos entre ellas para valores bajos de arcillas (0-10%), en especial los cambios abruptos que experimentan x_1 , x_2 , x_4 y x_5 para valores bajos.

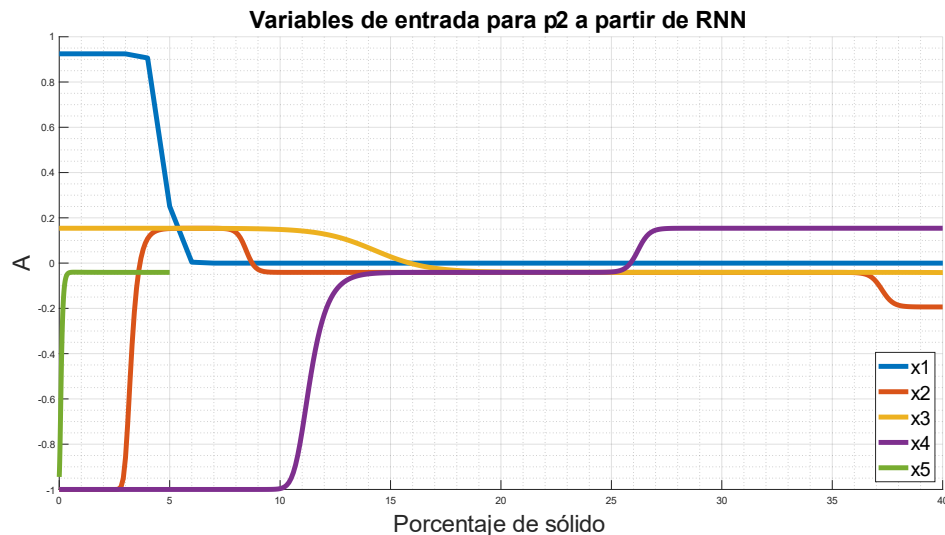


Figura 33: Variables involucradas en la modelación de p_2 a partir de RNN.

En la matriz w_{p_2} se presentan los pesos correspondientes a cada neurona de cada variable de entrada x . Se observa para este caso que existen pesos que se alejan de la tendencia general, en especial en x_2 y x_4 , lo que muestra un sobreajuste en la modelación. Esto se debe principalmente a la poca cantidad de datos disponibles para entrenar a la red, junto a la distribución de estos datos, que se comentó en la sección 5.1.

$$w_{p_2} = \begin{pmatrix} 1.2445 & 3.6613 & 2.0686 & -6.6495 & 1.7324 \\ -0.0061 & -1.0740 & 0.3679 & -0.2488 & -0.1687 \\ -0.2303 & 0.9427 & 0.1314 & -1.3474 & 0.0840 \\ -0.2448 & 0.9378 & 0.1169 & -1.3556 & 0.0831 \\ 0.1494 & 1.4859 & 0.7260 & -6.4946 & 0.1653 \\ -4.2580 & -6.6158 & 0.9205 & 2.5761 & 0.7477 \\ 1.3529 & 21.9266 & 7.6323 & -27.1733 & 2.1106 \\ 3.5380 & 5.1384 & 6.7637 & -16.9116 & 8.4392 \\ 2.6107 & 4.7500 & 5.1517 & -12.7276 & 0.1555 \\ 0.8421 & 12.4194 & 4.7847 & -16.7546 & 1.9856 \end{pmatrix}$$

En la Figura 34 se observa el efecto de cada variable de x sobre el coeficiente p_3 . Se puede ver que las variables presentan un comportamiento cóncavo en distintos tramos. La variable x_1 se presenta en la línea constante tapada por las demás curvas, siendo ésta la que domina el comportamiento de p_3 .

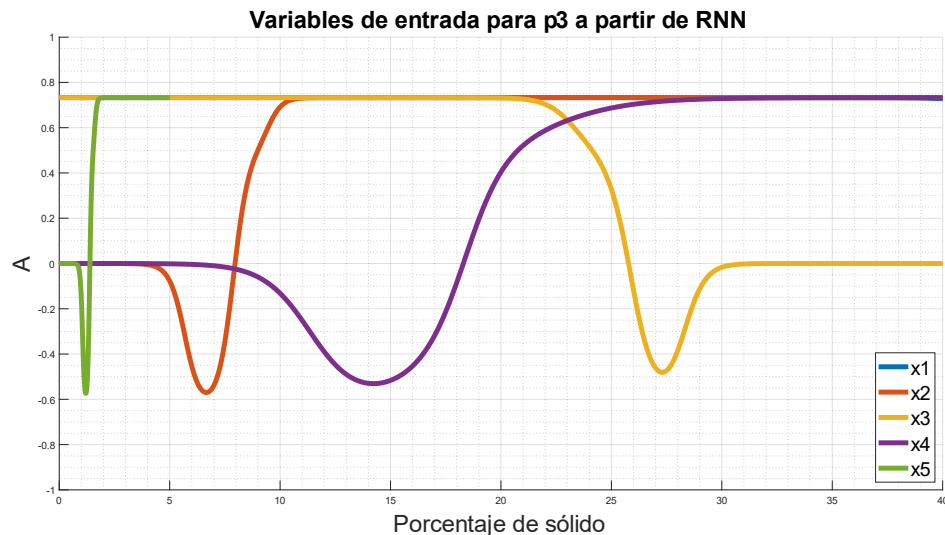


Figura 34: Variables involucradas en la modelación de p_3 a partir de RNN.

En la matriz w_{p_3} se presentan los pesos correspondientes a cada neurona de cada variable de entrada x . Se observa para este caso que los valores son similares, sin pesos que se escapen de la tendencia general. Las variables x_1 , x_2 y x_5 presentan mayoritariamente valores positivos, mostrando una relación directa con p_3 .

$$w_{p_3} = \begin{pmatrix} 2.3866 & 2.3646 & -0.9821 & -2.8065 & 1.6790 \\ 0.7523 & 0.4869 & -0.7804 & -1.9767 & 1.6776 \\ 0.8700 & 1.4935 & -0.4438 & -1.2296 & 1.8127 \\ 0.4199 & 0.1875 & 0.2328 & 0.0603 & 1.8160 \\ 0.5276 & 0.2627 & -0.0038 & -0.1569 & 1.8042 \\ 4.6017 & -8.2651 & 6.5054 & -9.4704 & 3.4465 \\ 0.7178 & -0.6853 & -0.3407 & -0.8720 & 1.7569 \\ -0.0528 & -0.9930 & 1.2166 & 0.6602 & 1.7646 \\ -0.0223 & 0.4972 & -0.5792 & 0.4073 & 1.7810 \\ 0.7556 & 0.2939 & 0.2706 & 0.1143 & 1.8224 \end{pmatrix}$$

En la Figura 35 se observa el efecto de cada variable de x sobre el coeficiente ϕ_c . Se puede ver que las variables convergen en los valores medios del rango de aplicación, aunque x_1 corresponde a la variable que se aleja más de la tendencia en los valores bajos de x_1 .

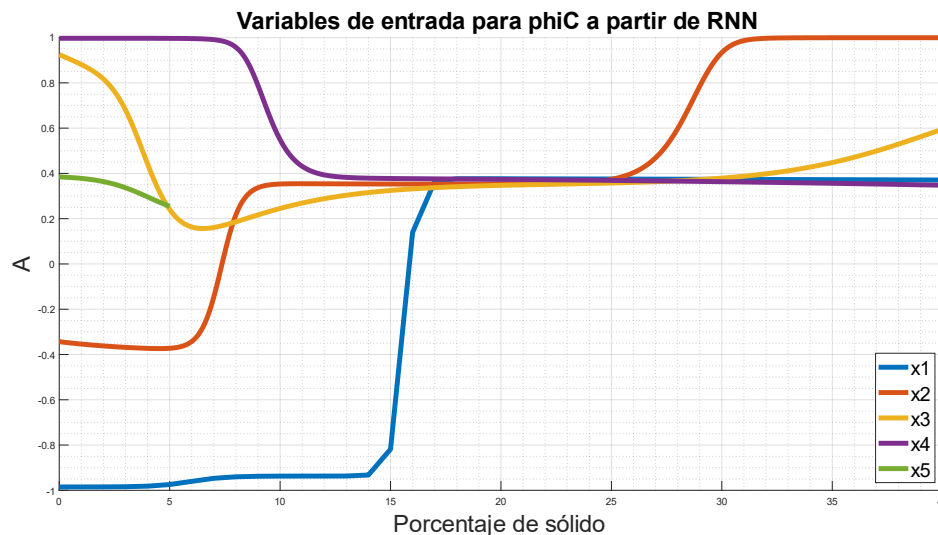


Figura 35: Variables involucradas en la modelación de ϕ_c a partir de RNN.

En la matriz w_{ϕ_c} se presentan los pesos correspondientes a cada neurona de cada variable de entrada x . Se observa para este caso que los valores son similares, sin pesos que se escapen de la tendencia general. Las variables x_2 y x_5 presentan mayoritariamente valores positivos, mostrando una relación directa con ϕ_c , mientras que las otras muestran principalmente valores negativos, es decir, una relación inversa con ϕ_c .

$$w_{\phi_c} = \begin{pmatrix} 0.6739 & -0.4356 & 0.1674 & -0.5374 & 1.0165 \\ -0.3024 & 0.4735 & -0.4518 & -0.1472 & 0.9505 \\ -0.3662 & 0.0932 & -0.1886 & -0.2709 & 0.9435 \\ -0.3319 & 0.3624 & 0.4453 & -0.7440 & 1.1719 \\ -0.3002 & 0.4631 & -0.4522 & -0.1360 & 0.9513 \\ -0.0055 & 1.2772 & -5.0704 & -0.4952 & -0.0685 \\ 0.8030 & 0.4643 & -0.5759 & -0.4786 & 0.9795 \\ 0.6179 & 1.7193 & -0.7401 & -0.2827 & 1.0935 \\ -0.3428 & -0.6348 & -0.1935 & 0.4731 & 0.9827 \\ 0.5533 & -0.1675 & 0.2944 & -0.3982 & 1.0525 \end{pmatrix}$$

Redes neuronales convolucionales

En la Figura 36 se observa el efecto de cada variable de x sobre el coeficiente A . Se puede ver que las variables presentan un rango similar en cuanto a su efecto en A , siendo x_4 la variable que presentan más cambios en su curva.

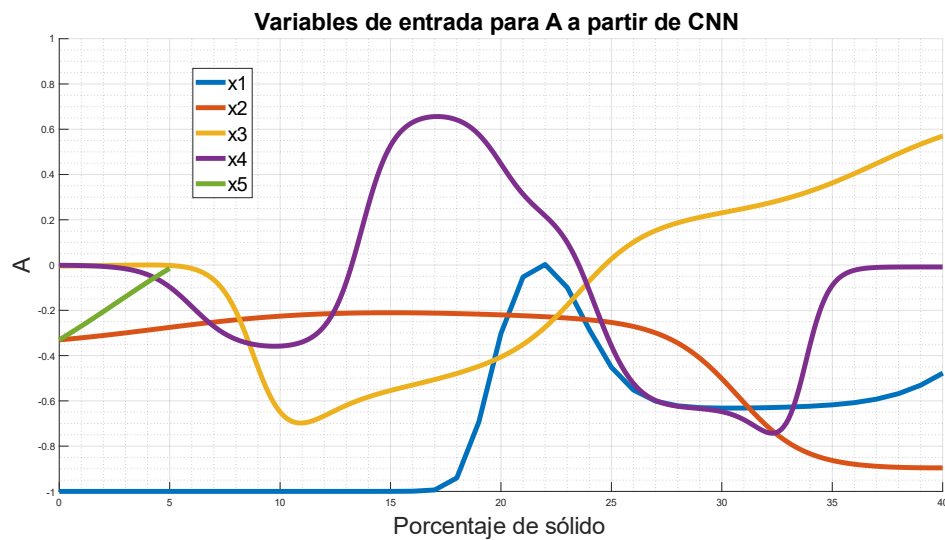


Figura 36: Variables involucradas en la modelación de A a partir de CNN.

En la matriz w_A se presentan los pesos correspondientes a cada neurona de cada variable de entrada x . Se observa para este caso que los valores son similares, sin pesos que se escapan de la tendencia general. Las variables x_2 , x_4 y x_5 presentan mayoritariamente valores positivos, mostrando una

relación directa con A , mientras que las otras muestran principalmente valores negativos, es decir, una relación inversa con A .

$$w_A = \begin{pmatrix} 0.2018 & 0.3309 & -0.8832 & 1.3749 & 0.3460 \\ -1.3771 & -0.2829 & -0.1327 & 1.4533 & 0.3590 \\ -0.4594 & 0.0419 & -0.7953 & 1.1241 & 0.3380 \\ -0.4635 & 0.2709 & -0.6047 & 1.2961 & 0.3388 \\ -0.5930 & 0.1042 & -0.0655 & 1.2235 & 0.3581 \\ -0.0954 & 0.7718 & -0.7418 & 0.9177 & 0.2941 \\ -0.1400 & -0.1157 & -1.1779 & 0.9995 & 0.3460 \\ -0.2024 & 0.2051 & -0.6690 & 1.2923 & 0.3414 \\ -0.3246 & -0.1614 & -0.6781 & 1.1668 & 0.2909 \\ -0.4664 & 0.3122 & -0.6009 & 1.2756 & 0.3370 \end{pmatrix}$$

En la Figura 37 se observa el efecto de cada variable de x sobre el coeficiente b . Se observa que a nivel general las variables presentan la misma relevancia sobre A excepto x_5 que presenta un rango en A mucho menor y alejado de las demás variables. Por otro lado, la variable x_1 (% de caolinita) es la que presenta el rango más amplio.

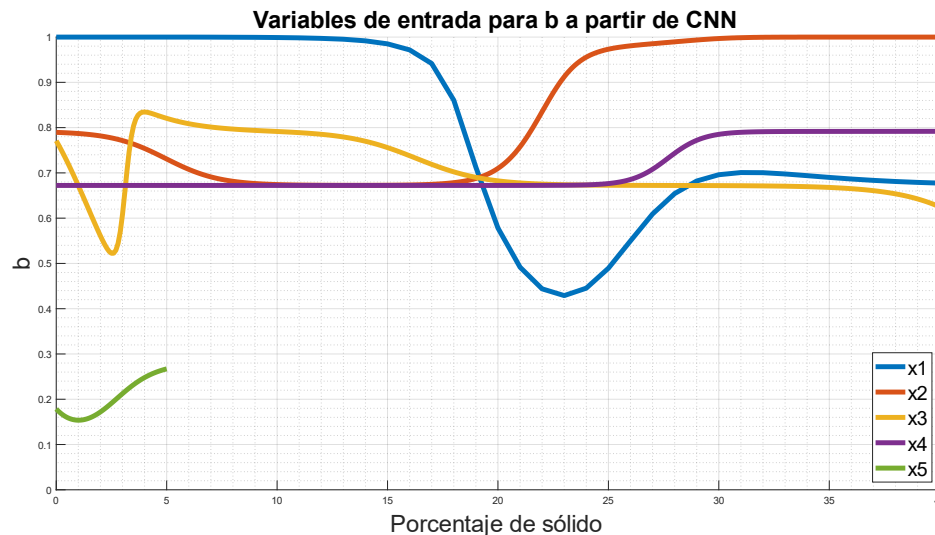


Figura 37: Variables involucradas en la modelación de b a partir de CNN.

En la matriz w_b se presentan los pesos correspondientes a cada neurona de cada variable de entrada x . Se observa en este caso que las variables x_2 y x_4 presenta valores positivos para sus pesos correspondientes, mostrando una

relación directa con b , mientras que para x_1 , x_3 y x_5 (% de caolinita y esmectita, respectivamente) se observan mayoritariamente valores negativos, reflejando una relación inversa con el coeficiente b .

$$w_b = \begin{pmatrix} -0.8495 & -0.0104 & 0.5253 & 0.2149 & -0.1969 \\ -0.3389 & 0.1181 & -0.8712 & 1.0613 & -0.2065 \\ -0.0853 & 0.1811 & -0.4010 & 0.2743 & -0.2055 \\ -0.4040 & 0.3743 & -0.1675 & 0.2928 & -0.2234 \\ -0.3932 & 0.4088 & -0.1736 & 0.2522 & -0.2247 \\ -0.4495 & 0.4045 & -0.1018 & 0.1728 & -0.2277 \\ -0.3558 & 0.3942 & -0.4446 & 0.8027 & -0.2035 \\ 0.0394 & -0.0362 & -0.3333 & 0.7752 & -0.1915 \\ -0.4330 & 0.4032 & -0.1150 & 0.1848 & -0.2282 \\ 0.2112 & 0.4327 & -0.1397 & -0.5166 & -0.1490 \end{pmatrix}$$

En la Figura 38 se observa el efecto de cada variable de x sobre el coeficiente C . Se puede ver que las variables x_1 y x_3 (% de caolinita y esmectita, respectivamente) presentan un rango mayor al de las otras variables, distanciándose de ellas en especial para valores bajos de x_1 (0-15%).

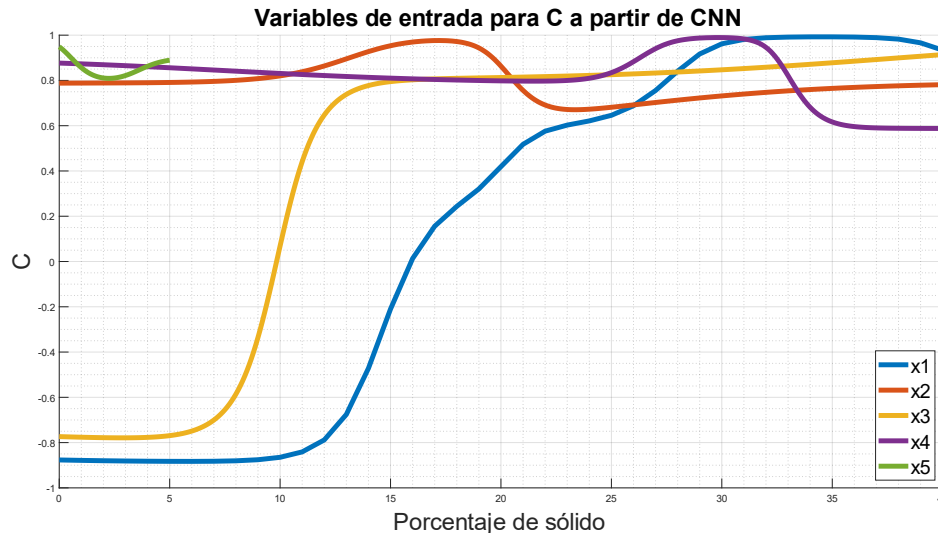


Figura 38: Variables involucradas en la modelación de C a partir de CNN.

En la matriz w_C se presentan los pesos correspondientes a cada neurona de cada variable de entrada x . Se observa en este caso que la variable x_5 presenta valores estrictamente positivos para sus pesos correspondientes,

mostrando una relación directa entre x_5 y C , mientras que para x_1 y x_4 (% de caolinita y moscovita, respectivamente) se observan mayoritariamente valores negativos, reflejando una relación inversa con el coeficiente C .

$$w_C = \begin{pmatrix} -0.0577 & -0.0184 & -0.4409 & -0.2206 & 0.8154 \\ 0.7371 & 0.0172 & -0.1324 & -0.2042 & 0.7998 \\ -0.0788 & -0.0157 & -0.0859 & -0.1499 & 0.7986 \\ -0.0001 & -0.0144 & -0.2660 & -0.3571 & 0.7857 \\ -0.0857 & -0.0155 & -0.3191 & -0.2027 & 0.7983 \\ -0.2566 & -0.3725 & 0.3148 & -0.2366 & 0.7943 \\ -0.0809 & 0.9823 & -0.2733 & -0.1470 & 0.8445 \\ -0.0524 & -0.0189 & 0.0076 & -1.4483 & 0.8656 \\ 1.2136 & 0.1622 & -0.7501 & -0.2172 & 0.8073 \\ 1.5069 & 0.1161 & 0.3625 & -0.7789 & 0.9324 \end{pmatrix}$$

En la Figura 39 se observa el efecto de cada variable de x sobre el coeficiente d . Se puede ver que las variables x_2 y x_3 (% de clorita y esmectita, respectivamente) presenta un rango mayor al de las otras variables, distanciándose de ellas en especial para valores bajos en el caso de x_2 (0-10%) y valores altos para x_3 (30-40%).

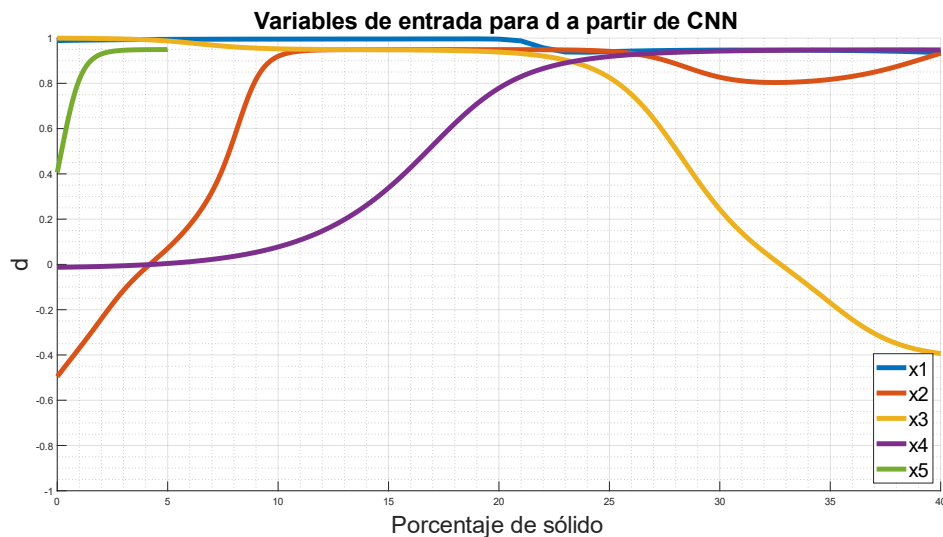


Figura 39: Variables involucradas en la modelación de d a partir de CNN.

En la matriz w_d se presentan los pesos correspondientes a cada neurona de cada variable de entrada x . Se observa en este caso que las variables x_2 , x_4 y x_5 presentan mayoritariamente valores positivos para sus pesos correspondientes, mostrando una relación directa con d .

$$w_d = \begin{pmatrix} -0.2545 & 0.8467 & -0.7577 & 0.0868 & 0.2366 \\ -0.2564 & 0.8159 & -0.7705 & 0.0901 & 0.2369 \\ -0.1623 & 0.0742 & -0.2087 & -0.0888 & 0.2083 \\ -1.1591 & -0.3887 & 0.4227 & 0.7362 & 0.2095 \\ -0.3254 & 2.1550 & -0.4862 & 0.3210 & 0.3412 \\ -0.2735 & 0.8328 & -0.6919 & 0.1171 & 0.2346 \\ -0.8799 & 0.1458 & 0.3423 & 0.2318 & 0.2222 \\ -0.2766 & 0.1177 & -0.3739 & 0.0931 & 0.2187 \\ -0.5803 & -0.1233 & 1.1435 & -0.4729 & 0.2750 \\ -0.4283 & 0.3488 & 0.1491 & 0.1479 & 0.2288 \end{pmatrix}$$

En la Figura 40 se observa el efecto de cada variable de x sobre el coeficiente p_1 . Se puede ver que todas las variables presentan un rango similar, por lo que todas las variables tienen la misma relevancia sobre p_1 .

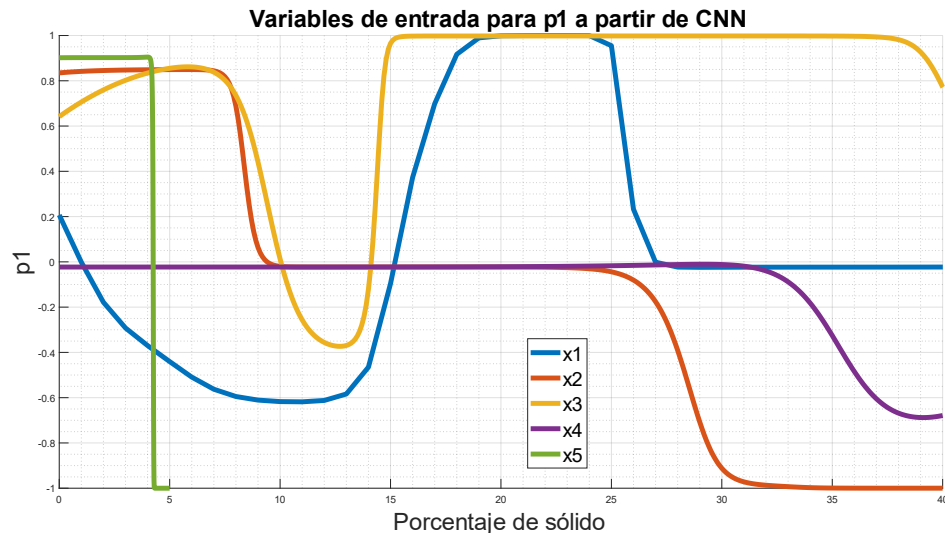


Figura 40: Variables involucradas en la modelación de p_1 a partir de CNN.

En la matriz w_{p_1} se presentan los pesos correspondientes a cada neurona de cada variable de entrada x . Se observa en este caso que la variable x_2 y x_5 presenta mayoritariamente valores negativos, mostrando una relación

inversa con p_1 . La matriz muestra valores similares en magnitud entre los pesos, mostrando un balance entre las neuronas de una misma variable.

$$w_{p_1} = \begin{pmatrix} -0.2662 & 0.0266 & 0.1715 & -0.4417 & -1.0065 \\ 0.5909 & -1.3636 & -0.9133 & 1.0135 & -0.9075 \\ 0.9502 & -1.2319 & -0.1061 & 1.6581 & -0.9718 \\ 0.5227 & 0.1872 & -0.5845 & 0.7356 & -1.0349 \\ -0.1255 & 0.3944 & 0.0382 & -0.5445 & -1.0151 \\ 0.1758 & -0.3077 & 0.6052 & 0.6352 & -0.9847 \\ -0.2532 & 0.0693 & 0.0016 & -0.4635 & -1.0225 \\ 0.2737 & 0.9449 & 1.0209 & -1.1715 & -0.9665 \\ 0.3183 & -0.0715 & -0.2142 & -0.4608 & -1.0262 \\ -0.6560 & -1.7732 & -3.1037 & 3.9421 & -2.7313 \end{pmatrix}$$

En la Figura 41 se observa el efecto de cada variable de x sobre el coeficiente p_2 . Se puede ver que las variables presentan comportamientos distintos entre ellas para valores bajos de arcillas (0-10%), en especial los cambios abruptos que experimentan x_1, x_2, x_4 y x_5 para valores bajos. La variable x_3 presenta un rango muy acotado, al igual que x_5 .

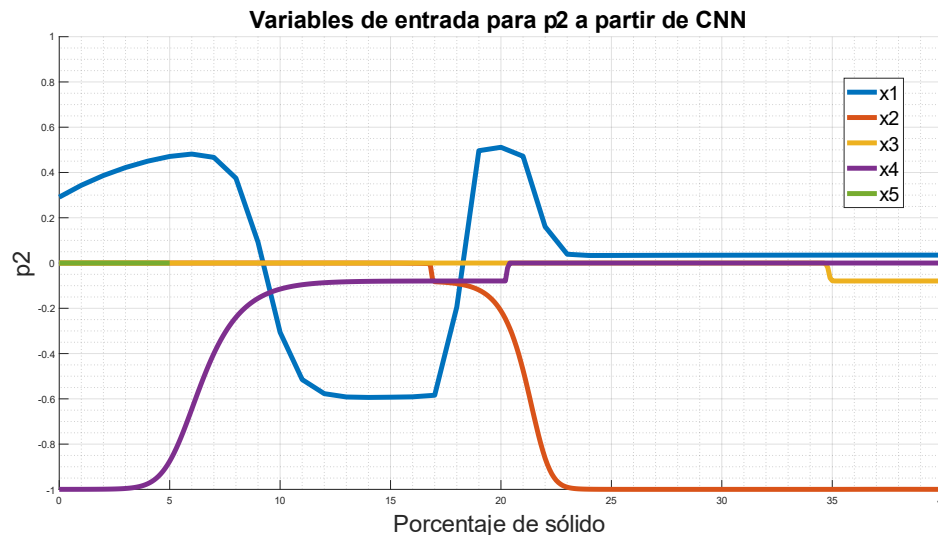


Figura 41: Variables involucradas en la modelación de p_2 a partir de CNN.

En la matriz w_{p_1} se presentan los pesos correspondientes a cada neurona de cada variable de entrada x . Se observa en este caso que las variables x_2 y x_5 presentan mayoritariamente valores negativos, mostrando una relación

inversa con p_1 . La matriz muestra valores que se alejan de la tendencia general para todas las variables, reflejando un sobreajuste para esta red neuronal. Esto al igual que en el caso de la red RNN se debe a la cantidad de datos experimentales y la distribución de éstos.

$$w_{p_2} = \begin{pmatrix} -0.4750 & 0.3355 & -0.2898 & -0.6520 & -0.3570 \\ -13.5975 & 45.9176 & 26.0644 & -77.2023 & -36.1633 \\ -0.4734 & 0.3285 & -0.2900 & -0.6503 & -0.3568 \\ -0.4656 & 0.2976 & -0.2909 & -0.6447 & -0.3559 \\ -0.6271 & 1.2181 & -0.0743 & -0.8662 & -0.3809 \\ -0.4387 & 0.2324 & -0.3155 & -0.6646 & -0.3559 \\ -0.2956 & 1.2347 & 0.1784 & -0.7735 & -0.3482 \\ -0.4696 & 0.3131 & -0.2904 & -0.6471 & -0.3563 \\ -2.1531 & 2.0930 & -0.3167 & 1.2824 & -3.7386 \\ -0.5338 & 0.3935 & -0.7786 & -0.9401 & -0.3766 \end{pmatrix}$$

En la Figura 42 se observa el efecto de cada variable de x sobre el coeficiente p_3 . Se puede ver que las variables presentan un rango similar, a excepción del acotado rango de x_5 y que para valores bajos la variable x_1 se distancia de las demás variables.

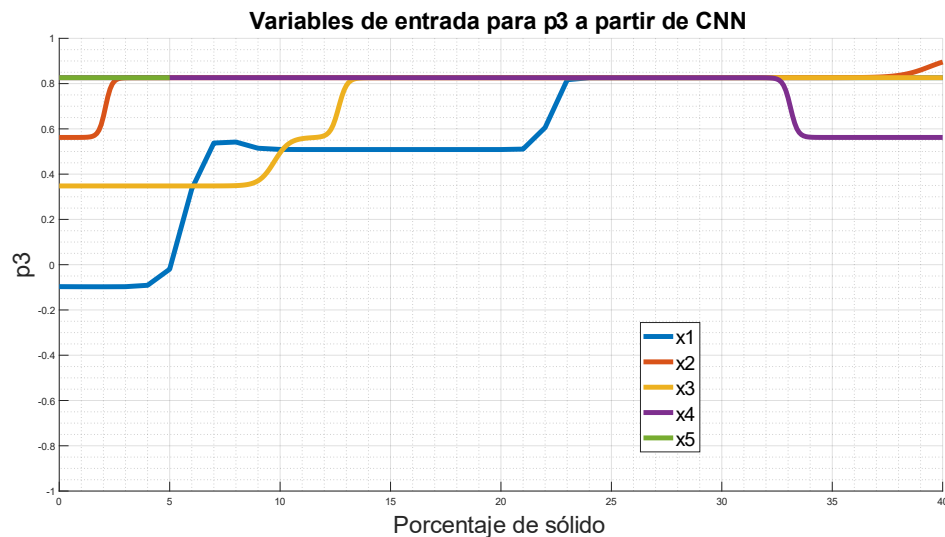


Figura 42: Variables involucradas en la modelación de p_3 a partir de CNN.

En la matriz w_{p_3} se presentan los pesos correspondientes a cada neurona de cada variable de entrada x . Se observa para este caso que los valores son

similares, sin pesos que se escapen de la tendencia general. Las variables x_1 , x_2 y x_5 presentan mayoritariamente valores positivos, mostrando una relación directa con p_3 .

$$w_{p_3} = \begin{pmatrix} 0.2892 & 6.7079 & 5.2865 & -6.0216 & 0.7505 \\ 2.6567 & 0.4356 & -1.0270 & -0.1552 & 0.5131 \\ -1.4469 & -0.4373 & -0.9093 & -0.2856 & 0.4179 \\ -0.3217 & -1.0992 & -0.1745 & -1.5141 & 0.4422 \\ 4.2506 & 0.6578 & -3.9581 & -0.5261 & 0.4582 \\ 0.9009 & -1.5169 & 0.3814 & 0.4826 & 0.5271 \\ 2.5582 & 1.5337 & -1.2205 & -0.1375 & 0.4561 \\ 1.1493 & -1.0192 & -2.8882 & -0.4933 & 0.4676 \\ 1.3214 & 0.4260 & 0.9540 & 0.5187 & 0.5416 \\ -0.8024 & 0.1086 & -0.4888 & -0.9703 & 0.4358 \end{pmatrix}$$

En la Figura 35 se observa el efecto de cada variable de x sobre el coeficiente ϕ_c . Se puede ver que las variables convergen en los valores medios del rango de aplicación, aunque x_4 corresponde a la variable que se aleja más de la tendencia en sus valores altos.

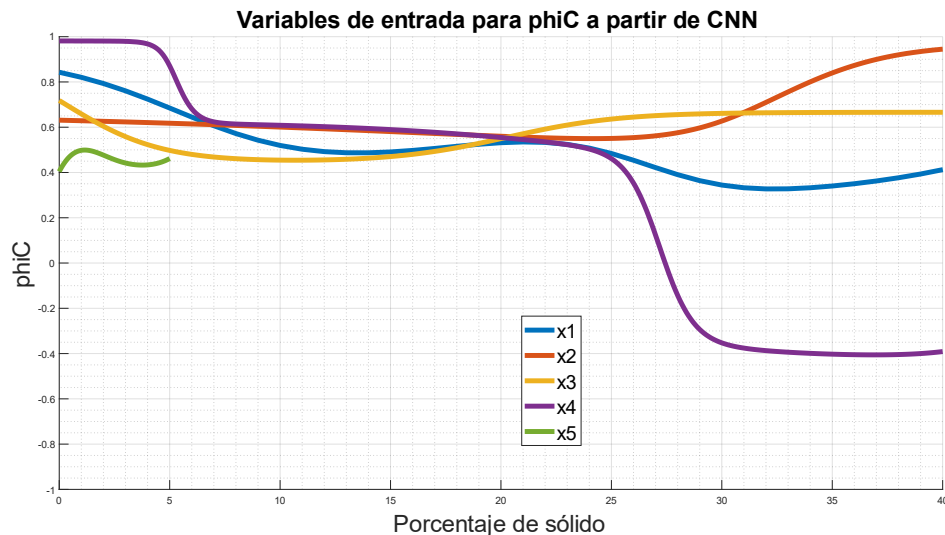


Figura 43: Variables involucradas en la modelación de ϕ_c a partir de CNN.

En la matriz w_{ϕ_c} se presentan los pesos correspondientes a cada neurona de cada variable de entrada x . Se observa para este caso que los valores son similares, sin pesos que se escapen de la tendencia general. Las variables x_2

y x_5 presentan mayoritariamente valores positivos, mostrando una relación directa con ϕ_c , mientras que las otras muestran principalmente valores negativos, es decir, una relación inversa con ϕ_c .

$$w_{\phi_c} = \begin{pmatrix} -0.0558 & 0.4465 & -0.4409 & -0.2206 & 0.8154 \\ -0.0788 & 0.0569 & -0.1324 & -0.2042 & 0.7998 \\ -0.0933 & 0.0057 & -0.0859 & -0.1499 & 0.7986 \\ -0.0522 & 0.0581 & -0.2660 & -0.3571 & 0.7857 \\ -0.0703 & 0.3477 & -0.3191 & -0.2027 & 0.7983 \\ -0.0059 & 0.4660 & -0.4638 & -0.2366 & 0.7943 \\ -0.1174 & -0.0800 & 0.3875 & -0.1470 & 0.8445 \\ -0.3806 & 0.3020 & 0.8475 & -1.4483 & 0.8656 \\ -0.0597 & 0.4604 & -0.4470 & -0.2172 & 0.8073 \\ 0.7102 & -0.4143 & 0.0664 & -0.7789 & 0.9324 \end{pmatrix}$$

5.4 Resultados obtenidos por los modelos de Machine Learning

A continuación, se muestran distintas tablas que muestran los resultados obtenidos a partir de los modelos de Machine Learning diseñados, comparados con los valores de los datos obtenidos. Cabe destacar que estos valores se presentan normalizados en función del valor máximo de cada serie de datos, esto debido a que los valores originales no pueden ser mostrados por temas de confidencialidad. En el Anexo B se pueden observar gráficos que representan esta información.

Función batch de Kynch

En las Tablas 7 y 8 se muestran los resultados obtenidos para los coeficientes a estimar de la función batch de Kynch, presentada en la ecuación 3.4.1. Para ambos coeficientes se observan buenos resultados, exceptuando la predicción de algunos valores para el caso del algoritmo de programación genética, que se alejan del valor deseado, como es el caso de las muestras R05 y R06 para el coeficiente A , que corresponden a muestras con un valor bajo para dicho coeficiente. En la sección 5.6 se muestran los errores de los modelos con respecto a cada uno de los coeficientes.

Tabla 7: Valores normalizados obtenidos por cada modelo. Pre-exponencial función batch de Kynch.

Muestra	A	A_{GP}	A_{RNN}	A_{CNN}
R01	-1	-0,9192	-0,9964	-0,9929
R02	-0,0082	-0,0422	-0,0087	-0,0087
R03	-0,6239	-0,7249	-0,6227	-0,6230
R04	-0,8037	-0,7186	-0,8179	-0,8170
R05	-0,0040	-0,1166	-0,0040	-0,0118
R06	-0,0288	-0,1316	-0,0293	-0,0411
R07	-0,5870	-0,5931	-0,5882	-0,5842
R08	-0,3743	-0,0422	-0,3743	-0,3743
R09	-0,8373	-0,6639	-0,8368	-0,8349
R10	-0,0738	-0,0707	-0,0737	-0,0798
R11	-0,0402	-0,0381	-0,0411	-0,0395
R12	-0,6046	-0,5438	-0,6045	-0,6059
R13	-0,8255	-0,7221	-0,8214	-0,8261
R14	-0,0979	-0,1241	-0,0996	-0,0764
R15	-0,1505	-0,0539	-0,0087	-0,1463

Tabla 8: Valores normalizados obtenidos por cada modelo. Exponente función batch de Kynch.

Muestra	b	b_{GP}	b_{RNN}	b_{CNN}
R01	0,7764	0,5344	0,7999	0,7521
R02	0,2517	0,2150	0,2243	0,2605
R03	0,7102	0,6738	0,6979	0,6899
R04	0,7338	0,6735	0,7039	0,7020
R05	0,2053	0,2474	0,2396	0,2273
R06	0,3482	0,4217	0,3552	0,3259
R07	0,7292	0,6384	0,7258	0,6976
R08	0,6720	0,6995	0,6731	0,6667
R09	0,8375	0,7333	0,8889	0,7816
R10	0,1628	0,1617	0,1561	0,1720
R11	1	1,0439	0,9998	0,9938
R12	0,4926	0,6262	0,5031	0,4785

R13	0,5406	0,6737	0,6008	0,5833
R14	0,3387	0,3509	0,3334	0,3133
R15	0,4336	0,4847	0,4335	0,3808

En la Figura 44 y 45 se observan los gráficos de valores experimentales versus valores predichos por los distintos modelos de Machine Learning. Para el caso de la variable A, el modelo que más se aproxima a la línea diagonal (coincidencia del valor empírico y del modelo) es la red neuronal convolucional, mientras que el modelo de programación genética se distancia de ésta. Por otra parte, para el caso de la variable b, se conserva esta tendencia, siendo la red neuronal convolucional la que presenta más coincidencias, y el modelo de programación genética el más lejano a la recta.

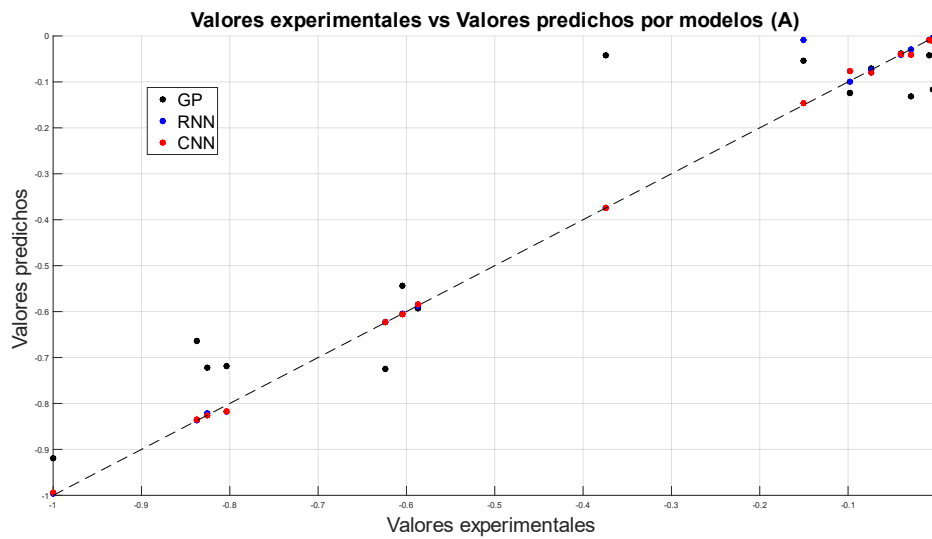


Figura 44: Valores experimentales y predichos para A.

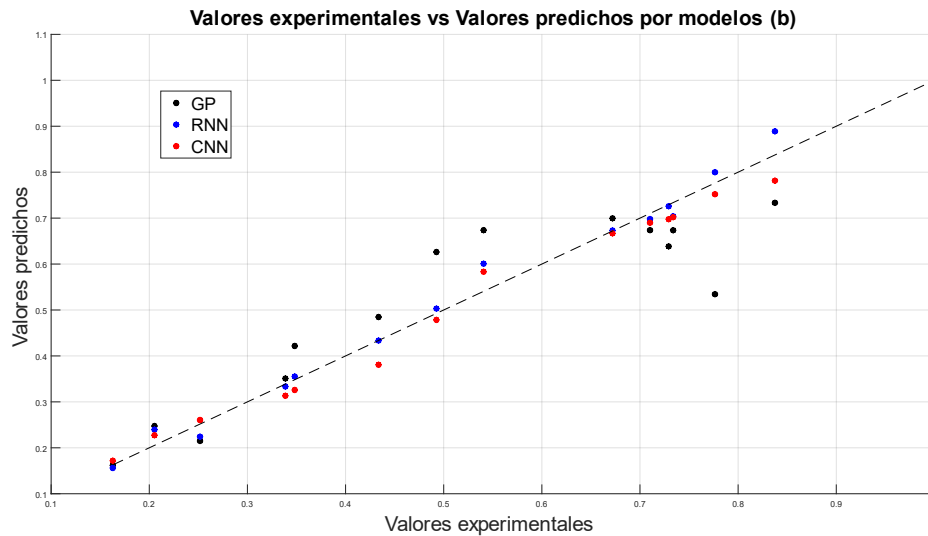


Figura 45: Valores experimentales y predichos para b.

Función estrés efectivo de sólido

En las Tablas 9 y 10 se muestran los resultados obtenidos para los coeficientes a estimar de la función estrés efectivo del sólido, presentada en la ecuación 3.4.2. Para ambos coeficientes se observan que las predicciones coinciden en buena forma con los resultados esperados, aunque sí existe un error mayor para el algoritmo diseñado a partir de programación genética. Se puede notar para estos casos que los resultados no se distancian mayormente para algunas muestras, como el caso de la función batch de Kynch, teniéndose similares para todas las muestras de relaves. En la sección 5.6 se muestran los errores de los modelos con respecto a cada uno de los coeficientes.

Tabla 9: Valores normalizados obtenidos por cada modelo. Pre-exponencial función estrés efectivo del sólido.

Muestra	C	C_{GP}	C_{RNN}	C_{CNN}
R01	0,6309	0,6873	0,6229	0,6379
R02	0,4776	0,4751	0,4761	0,4776
R03	0,7492	0,7512	0,7579	0,7507
R04	0,7424	0,7496	0,7421	0,7418
R05	1	0,8472	0,9964	0,9971

R06	0,6567	0,8208	0,6606	0,6571
R07	0,7971	0,7889	0,8045	0,7998
R08	0,5258	0,5163	0,5236	0,5259
R09	0,8071	0,8372	0,8112	0,8108
R10	0,8441	0,8355	0,8319	0,8378
R11	0,4793	0,4735	0,4794	0,4792
R12	0,6052	0,6089	0,6060	0,6052
R13	0,7501	0,7507	0,7513	0,7497
R14	0,8835	0,8367	0,8838	0,8835
R15	0,8180	0,7862	0,8179	0,8204

Tabla 10: Valores normalizados obtenidos por cada modelo. Exponente función estrés efectivo del sólido.

Muestra	d	d_{GP}	d_{RNN}	d_{CNN}
R01	0,8403	0,7774	0,8328	0,8311
R02	0,4429	0,4572	0,4403	0,4527
R03	0,6023	0,6546	0,5816	0,6022
R04	0,7587	0,7361	0,7606	0,7564
R05	0,4967	0,8467	0,4982	0,4967
R06	0,7910	0,8161	0,8005	0,7961
R07	0,8815	0,8425	0,8909	0,8788
R08	0,4881	0,4852	0,4921	0,4805
R09	1	0,9927	0,9999	0,9997
R10	0,9673	0,9464	0,9655	0,9756
R11	0,7875	0,8495	0,7981	0,7875
R12	0,9527	0,9052	0,9106	0,9507
R13	0,7771	0,7369	0,7766	0,7845
R14	0,9329	0,8319	0,8914	0,9328
R15	0,8199	0,8978	0,8232	0,8194

En las Figuras 46 y 47 se observan los gráficos de valores experimentales versus valores predichos por los distintos modelos de Machine Learning. Para el caso de la variable C, el modelo que más se aproxima a la línea diagonal (coincidencia del valor empírico y del modelo) es la red neuronal recurrente, mientras que el modelo de programación genética se distancia

de ésta. Por otra parte, para el caso de la variable d, la red neuronal convolucional presenta más coincidencias, mientras que el modelo de programación genética es el más lejano a la recta.

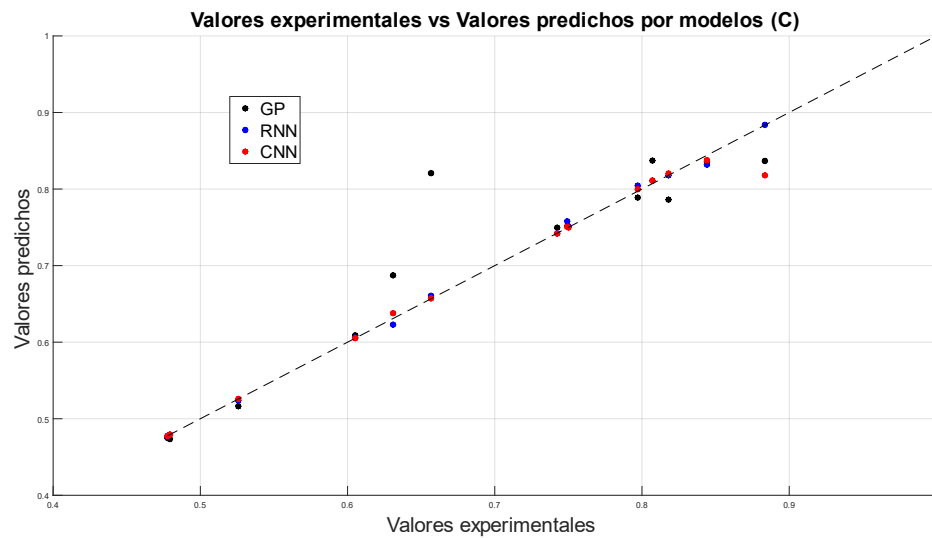


Figura 46: Valores experimentales y predichos para C.

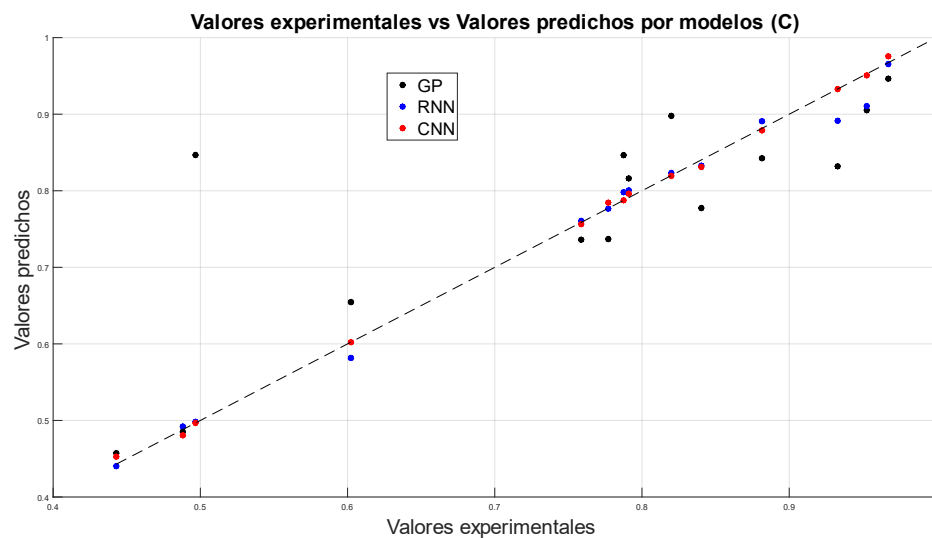


Figura 47: Valores experimentales y predichos para d.

Función de dosificación

En las Tablas 11, 12 y 13 se muestran los resultados obtenidos para los coeficientes a estimar de la función de dosificación, presentada en la ecuación 3.4.3. Para el coeficiente cuadrático p_1 y el coeficiente lineal p_2 se observan grandes discrepancias en cuanto a lo predicho por los modelos, en especial en el algoritmo de programación genética, donde el modelo no acierta en la tendencia de los datos. La muestra donde se observa en mayor medida este problema es R06 y R13, que además corresponden a las muestras donde estos coeficientes presentan el menor valor. Con respecto al coeficiente libre p_3 se observa una mejor predicción general de los datos, aunque nuevamente se ve un problema para simular las muestras con valores bajos en estos coeficientes. En la sección 5.6 se muestran los errores de los modelos con respecto a cada uno de los coeficientes.

Tabla 11: Valores normalizados obtenidos por cada modelo. Coeficiente cuadrático función de dosificación.

Muestra	p_1	p_{1GP}	p_{1RNN}	p_{1CNN}
R01	-0,6214	-0,3325	-0,5608	-0,7332
R02	0,4516	0,4217	0,4819	0,3316
R03	0,7185	0,4521	0,6417	0,6529
R04	1	0,5264	0,8299	0,8608
R05	0,3044	0,2838	0,3799	0,3691
R06	0,0144	0,1193	0,1029	0,0394
R07	-0,2936	-0,1617	-0,3525	-0,1736
R08	0,8215	0,7451	0,8971	0,8388
R09	0,2086	0,4201	0,2121	0,3344
R10	0,6325	0,4127	0,6593	0,6118
R11	0,5740	0,5957	0,6668	0,5743
R12	0,3412	0,2749	0,3269	0,3988
R13	0,0082	0,4895	0,2017	0,2069
R14	0,3017	0,2518	0,3701	0,3622
R15	0,4369	0,5198	0,7677	0,4279

Tabla 12: Valores normalizados obtenidos por cada modelo. Coeficiente lineal función de dosificación.

Muestra	p_2	p_{2GP}	p_{2RNN}	p_{2CNN}
R01	0,6030	0,3544	0,6004	0,5698
R02	-0,0132	-0,0433	-0,0045	0,0122
R03	-0,0744	-0,0357	-0,0564	-0,0273
R04	-0,3307	-0,0354	-0,3317	-0,1738
R05	-0,0674	-0,0354	0,0680	-0,0853
R06	0,0004	-0,0352	0,0006	-0,0006
R07	0,1518	-0,0456	0,0524	0,0143
R08	-0,1306	-0,0809	-0,1344	-0,0839
R09	-0,0446	-0,0432	-0,0456	-0,0430
R10	-0,0869	-0,0465	-0,0693	-0,0428
R11	-1	-0,9997	-0,9993	-0,9999
R12	-0,0089	-0,0357	-0,0212	-0,0951
R13	$-8,723 \cdot 10^{-6}$	-0,0356	-0,0576	-0,0369
R14	-0,1341	-0,0352	-0,1348	-0,1144
R15	-0,0373	-0,0520	-0,0435	-0,0427

Tabla 13: Valores normalizados obtenidos por cada modelo. Coeficiente libre función de dosificación.

Muestra	p_3	p_{3GP}	p_{3RNN}	p_{3CNN}
R01	-0,0941	-0,0565	0,0939	-0,0835
R02	0,0492	0,0934	0,0845	0,2349
R03	0,6387	0,4799	0,5933	0,4784
R04	1	0,4827	0,9145	0,7273
R05	0,7802	0,5598	0,7585	0,6346
R06	0,7649	0,5410	0,4830	0,6068
R07	0,6045	0,5259	0,6181	0,5388
R08	0,7024	0,4295	0,5548	0,5993
R09	0,7746	0,6437	0,7649	0,7364
R10	0,6935	0,6295	0,7053	0,7189
R11	0,7711	0,7767	0,7639	0,7561

R12	0,7711	0,6157	0,7831	0,8092
R13	0,0341	0,4817	0,1193	0,2378
R14	0,0173	0,5509	0,1876	0,3357
R15	0,0726	0,5621	0,0501	0,2815

En las Figuras 48, 49 y 50 se observan los gráficos de valores experimentales versus valores predichos por los distintos modelos de Machine Learning. Para el caso de la variable p_1 , todos los modelos presentan notorias diferencias entre algunos puntos experimentales y sus respectivas simulaciones. Sin embargo, los modelos de redes neuronales presentan en general una mayor coincidencia con la línea diagonal, siendo el modelo de programación genética el que más se aleja en varios puntos. Para el caso de la variable p_3 , se observa una zona densa de punto en el medio del gráfico, siendo el modelo a partir de red neuronal recurrente el que presenta puntos que se alejan en menor medida de la línea diagonal., mientras que el modelo de programación tiene puntos que se alejan bastante de esta tendencia. Por último, para el caso de la variable p_3 , las diferencias entre los valores experimentales y predichos son muy notorias, alejándose de la línea diagonal. Sin embargo, las redes neuronales presentan mayor cantidad de puntos con valores similares entre modelo y medición, mientras que el modelo de programación genética es la que presenta las mayores diferencias.

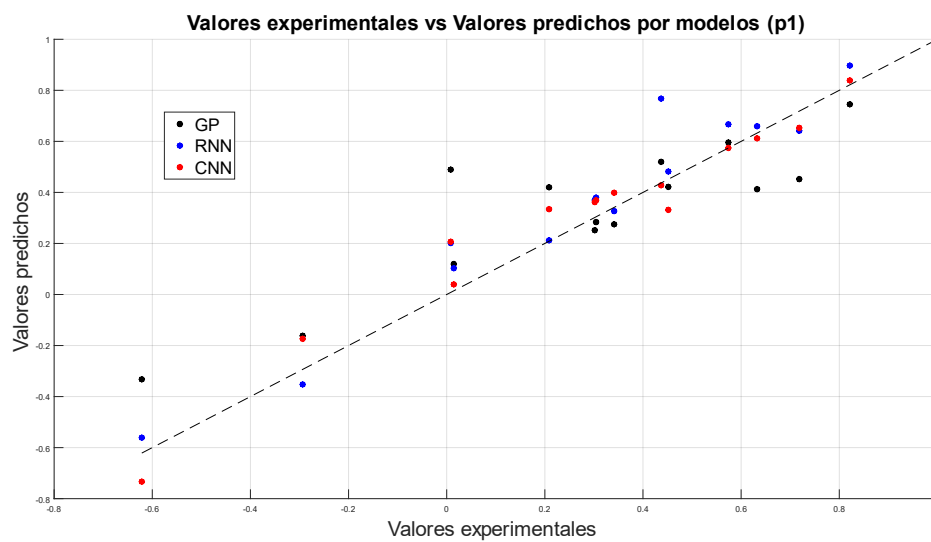


Figura 48: Valores experimentales y predichos para p_1 .

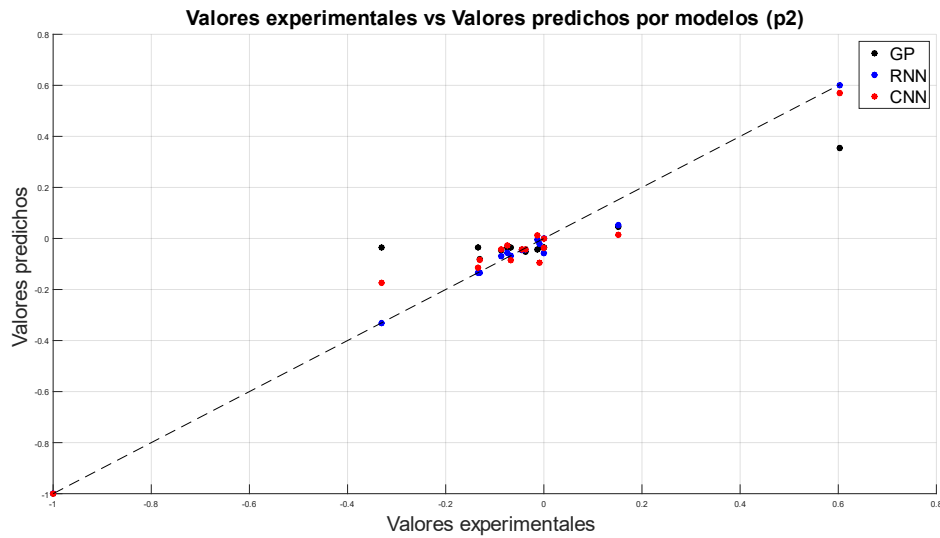


Figura 49: Valores experimentales y predichos para p_2 .

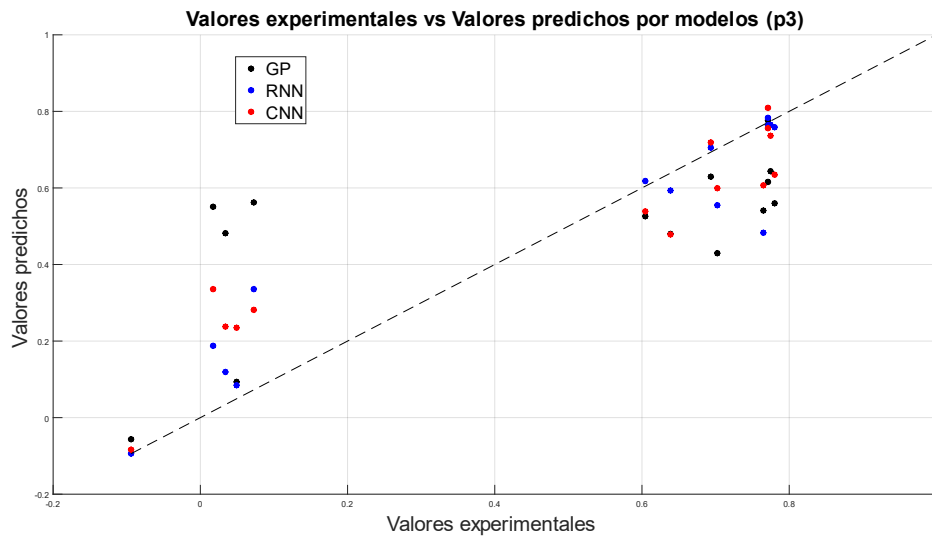


Figura 50: Valores experimentales y predichos para p_3 .

Concentración crítica

En la Tabla 14 se muestran los resultados obtenidos de la estimación de la concentración crítica, presentada en la ecuación 3.4.4. En términos generales se observa que todos los modelos presentan valores similares a

los expuestos por los datos experimentales. Sin embargo, el algoritmo de programación genética presenta valores más alejados a los empíricos, aunque no con una conducta tan pronunciada como en los casos anteriores. En la sección 5.6 se muestran los errores de los modelos con respecto a cada uno de los coeficientes.

Tabla 14: Valores normalizados obtenidos por cada modelo. Concentración crítica.

Muestra	ϕ_c	ϕ_{cGP}	ϕ_{cRNN}	ϕ_{cCNN}
R01	0,3849	0,4738	0,3831	0,4028
R02	0,5702	0,5770	0,5838	0,5576
R03	0,2914	0,2796	0,3147	0,3244
R04	0,3216	0,3031	0,3208	0,3230
R05	0,7917	0,6834	0,7686	0,7420
R06	0,7764	0,6321	0,7552	0,7300
R07	0,3789	0,4508	0,3939	0,4056
R08	0,5622	0,5793	0,5543	0,5868
R09	1	0,8627	0,9970	0,9876
R10	0,6947	0,7130	0,7146	0,7424
R11	0,8119	0,8631	0,8157	0,8330
R12	0,3922	0,4927	0,3922	0,4011
R13	0,3069	0,2913	0,3100	0,3233
R14	0,4123	0,6579	0,4330	0,4557
R15	0,7364	0,5693	0,7240	0,6831

En la Figura 51 se observa el gráfico de valores experimentales versus valores predichos por los distintos modelos de Machine Learning. Para el caso de la variable ϕ_c , el modelo que más se aproxima a la línea diagonal es la red neuronal recurrente, mientras que el modelo de programación genética se distancia de ésta.

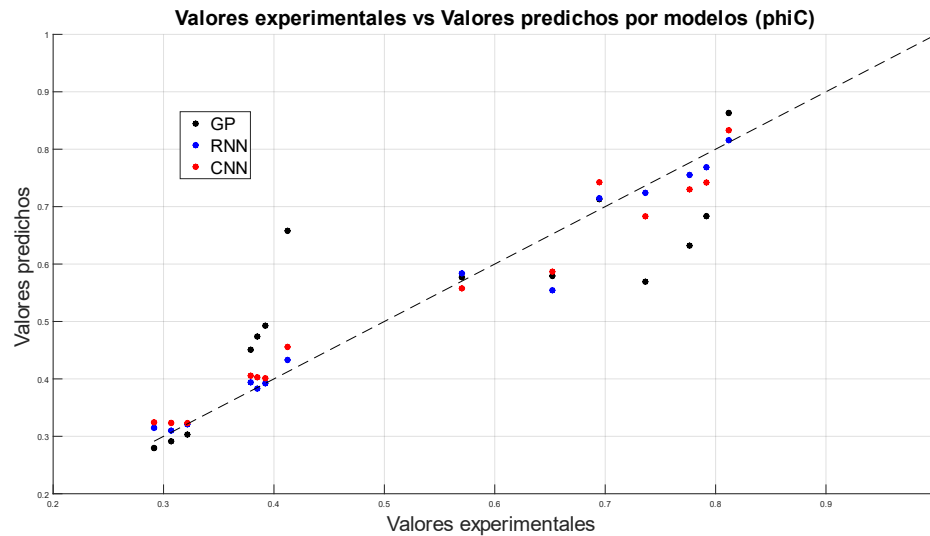


Figura 51: Valores experimentales y predichos para ϕ_c .

5.5 Comparación modelos: criterios estadísticos

Con respecto a los parámetros estadísticos para poder evaluar la efectividad de los modelos, en esta sección se muestra el error cuadrático medio ($RMSE$) y el coeficiente de correlación (R^2) para cada modelo con respecto a cada uno de los coeficientes de las funciones constitutivas.

Función batch de Kynch

Tabla 15: Coeficientes estadísticos obtenidos para cada modelo diseñado. Función batch de Kynch.

Modelo	A		b	
	R^2	$RMSE$	R^2	$RMSE$
GP	0,8979	$1,19 \cdot 10^{-1}$	0,8303	$1,02 \cdot 10^{-1}$
RNN	0,9890	$4,03 \cdot 10^{-3}$	0,9900	$2,58 \cdot 10^{-2}$
CNN	0,9955	$7,85 \cdot 10^{-3}$	0,9899	$2,92 \cdot 10^{-2}$

Se puede observar en la Tabla 15 que los coeficientes de correlación obtenidos por los modelos de redes neuronales (RNN y CNN) presentan valores muy cercanos a 1, teniendo a su vez los valores más bajos para el error cuadrático, tanto para A como para b . Esto muestra una gran relación

lineal entre los datos experimentales y los simulados. En cuanto al modelo de programación genética, se observan valores inferiores para el coeficiente de correlación, teniendo aproximadamente un error del 10%.

Función estrés efectivo de sólido

Tabla 16: Coeficientes estadísticos obtenidos para cada modelo diseñado. Función estrés efectivo del sólido.

Modelo	C		d	
	R^2	$RMSE$	R^2	$RMSE$
GP	0,8451	$7,43 \cdot 10^{-2}$	0,8594	$1,02 \cdot 10^{-1}$
RNN	0,9879	$5,14 \cdot 10^{-3}$	0,9917	$1,69 \cdot 10^{-2}$
CNN	0,9699	$2,91 \cdot 10^{-3}$	0,9910	$5,17 \cdot 10^{-3}$

En la Tabla 16 se muestran los parámetros estadísticos para los coeficientes de la función estrés efectivo del sólido. Se observan valores aceptables para todos los casos, destacando los valores de las redes neuronales, en especial la red neuronal recurrente (RNN), la cual presenta los valores más altos de R^2 para ambos coeficientes.

Función de dosificación

Tabla 17: Coeficientes estadísticos obtenidos para cada modelo diseñado. Función de dosificación.

Mod	p_1		p_2		p_3	
	R^2	$RMSE$	R^2	$RMSE$	R^2	$RMSE$
GP	0,7417	$2,24 \cdot 10^{-1}$	0,8634	$1,17 \cdot 10^{-1}$	0,3536	$2,88 \cdot 10^{-1}$
RNN	0,9259	$1,22 \cdot 10^{-1}$	0,9914	$3,06 \cdot 10^{-2}$	0,9210	$1,11 \cdot 10^{-1}$
CNN	0,9492	$9,41 \cdot 10^{-2}$	0,9591	$6,38 \cdot 10^{-2}$	0,8538	$1,60 \cdot 10^{-1}$

En la Tabla 17 se muestran los parámetros estadísticos para los coeficientes de la función de dosificación. Se observan valores aceptables para todos los casos, exceptuando para todos los modelos diseñados mediante programación genética. Esto se puede deber a los problemas de distribución

de los datos que presentaban las series de p_1 y p_2 , lo que a su vez perjudica el desempeño en el coeficiente p_3 ya que todas están conectadas.

Concentración crítica

Tabla 18: Coeficientes estadísticos obtenidos para cada modelo diseñado. Concentración crítica.

Modelo	ϕ_c	
	R^2	$RMSE$
GP	0,9694	$1,05 \cdot 10^{-1}$
RNN	0,9650	$1,41 \cdot 10^{-2}$
CNN	0,9821	$3,21 \cdot 10^{-2}$

En la Tabla 18 se muestran los parámetros estadísticos para la estimación de los modelos de la concentración crítica. Se observan valores aceptables para todos los casos, destacando desempeños similares para los 3 modelos en cuanto a los parámetros estadísticos.

5.6 Comparación de modelos

Función batch de Kynch: Efecto caolinita

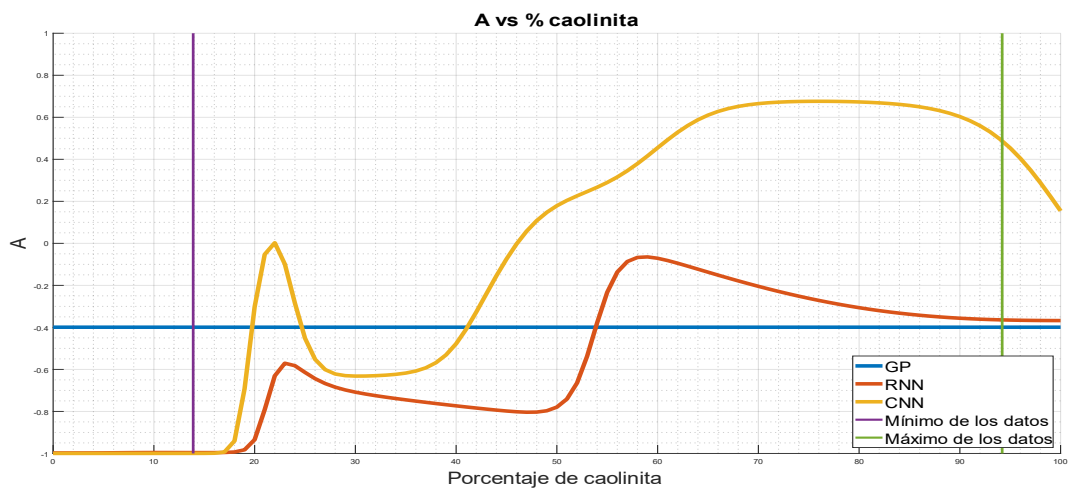


Figura 52: Simulación del coeficiente A en función del porcentaje de caolinita.

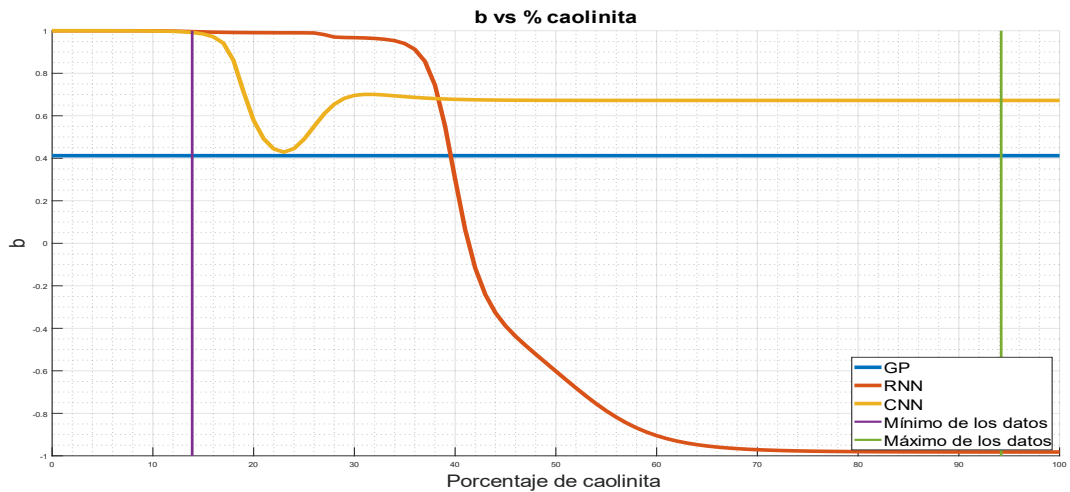


Figura 53: Simulación del coeficiente b en función del porcentaje de caolinita.

En las Figuras 52 y 53 se observa el comportamiento descrito por los modelos de Machine Learning con respecto al porcentaje de caolinita presente en la muestra para el caso de los coeficientes A y b . Se observa para ambos casos que el algoritmo de programación genética (GP) no logra captar la información de esta variable, manteniendo los coeficientes A y b constante al variar el porcentaje de caolinita. El modelo de red neuronal recurrente (RNN) para el caso del coeficiente A se muestran 2 tramos principales, donde el valor aumenta drásticamente aproximadamente al 20% de caolinita, y luego una paulatina disminución hasta llegar al 50% de caolinita; a continuación, la curva vuelve a crecer abruptamente, y terminando en un valor cercano al reportado por el modelo GP. Este se puede deber a que el modelo RNN logró identificar 2 subgrupos dentro de su conjunto de entrenamiento, para poder caracterizar de mejor manera este coeficiente. Este comportamiento se condice con lo esperado en cuanto a una disminución del coeficiente a medida que aumenta el porcentaje de este filosilicato, no obstante, no se tiene registro de subgrupos dentro de este comportamiento. Para el coeficiente b se observa que aproximadamente hasta el 35% de caolinita no afecta en gran medida a su valor, pero pasado este porcentaje la función disminuye exponencialmente de 1 hasta -1 , lo cual corresponde a un comportamiento esperado para este factor, debido a que las arcillas dificultan la sedimentación de las muestras, haciendo que ésta sea más lenta, es decir, con un valor de f_{bk} menor. Finalmente, para el modelo de red neuronal convolucional (CNN) se puede observar un

comportamiento sin un patrón definido para el caso del coeficiente A , alejándose considerablemente de los demás modelos, especialmente en los valores altos del porcentaje de caolinita. Para el coeficiente b se tiene una bajada brusca al principio de la curva, para luego volver a subir manteniéndose relativamente constante a partir del 30% de caolinita.

Con lo anterior, el modelo que mejor representa lo esperado para los coeficientes que componen la función de batch de Kynch es el modelo RNN, considerando que posee un comportamiento definido en todo el rango de aplicación y se condice con la teoría en cuanto a su efecto en la velocidad de sedimentación de la muestra.

Función batch de Kynch: Efecto clorita

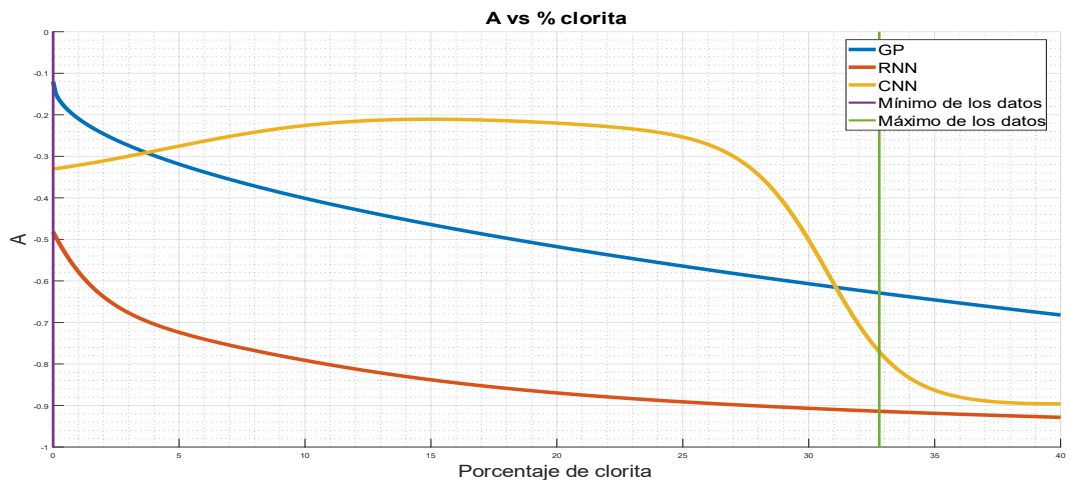


Figura 54: Simulación del coeficiente A en función del porcentaje de clorita.

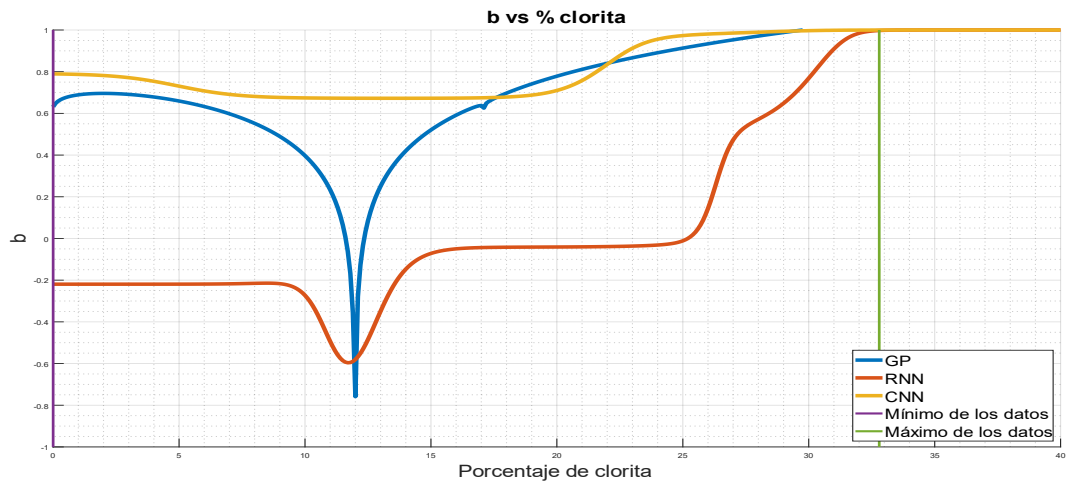


Figura 55: Simulación del coeficiente b en función del porcentaje de clorita.

En las Figuras 54 y 55 se observa el comportamiento descrito por los modelos de Machine Learning con respecto al porcentaje de clorita presente en la muestra para el caso de los coeficientes A y b . Para el caso del coeficiente A se observa que los modelos GP y RNN presentan una tendencia esperada para este caso, disminuyendo su valor a medida que aumenta el porcentaje de clorita, mientras que el modelo CNN no disminuye hasta el final de la curva, decreciendo abruptamente a un valor cercano al reportado por el modelo RNN. Por otra parte, para el coeficiente b , se observa una tendencia creciente para los 3 modelos. Sin embargo, el modelo GP presenta un comportamiento anómalo en torno al 12%, lo cual se puede deber a que el modelo primero se ajusta a un comportamiento decreciente y luego creciente en torno a este punto, aunque se esperaría una curva suave para este tipo de parámetros. El modelo RNN presenta un comportamiento mayoritariamente constante en gran parte de la curva, aumentando drásticamente a partir del 25% de clorita). Finalmente, para el modelo CNN, se observa un comportamiento constante hasta aproximadamente el 20%, aumentando levemente su valor a partir de este porcentaje.

Con lo anterior, el modelo que mejor representa lo esperado para los coeficientes que componen la función de batch de Kynch es el modelo RNN, ya que a pesar de presentar una tendencia creciente para el coeficiente b , esto lo puede compensar con el comportamiento que presenta el coeficiente A .

Función batch de Kynch: Efecto esmectita

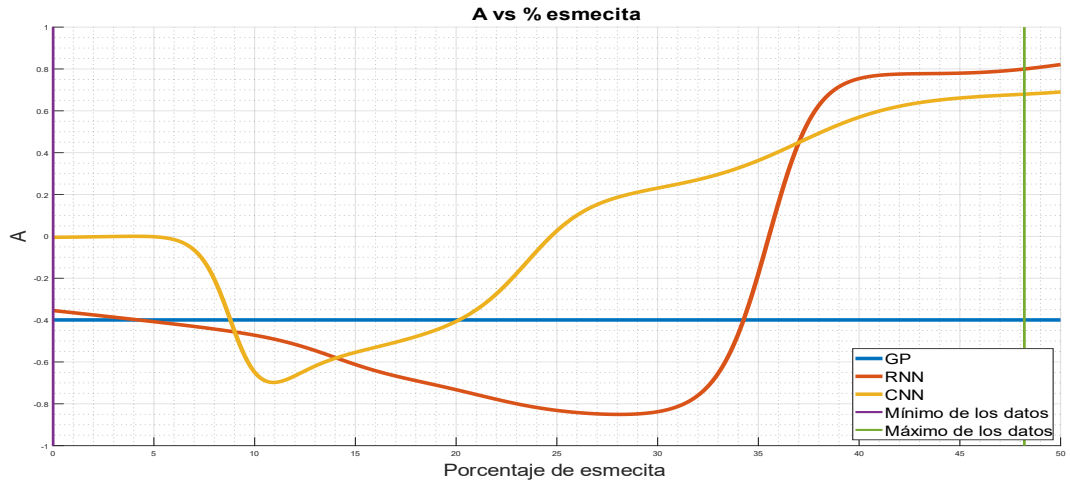


Figura 56: Simulación del coeficiente A en función del porcentaje de esmectita.

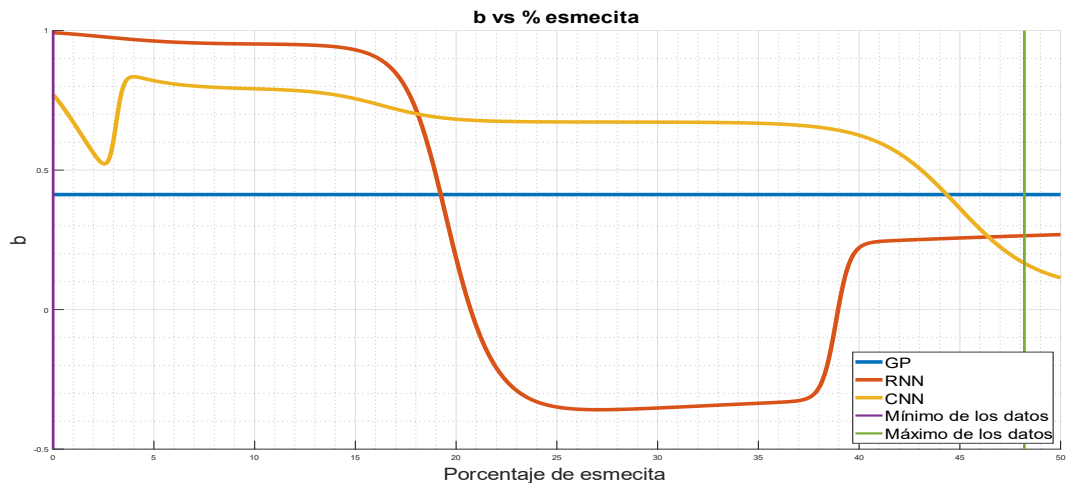


Figura 57: Simulación del coeficiente b en función del porcentaje de esmectita.

En las Figuras 56 y 57 se observa el comportamiento descrito por los modelos de Machine Learning con respecto al porcentaje de esmectita presente en la muestra para el caso de los coeficientes A y b . En el caso del modelo GP se observa un comportamiento similar a la caolinita para ambos coeficientes, ya que el modelo no se ve afectado por el porcentaje de esmectita, sino que se presenta un valor constante a pesar de que aumente la

variable independiente. Por otro lado, el modelo RNN para el coeficiente A presenta 2 zonas a grandes rasgos: primero, entre 0 y 30% aproximadamente, se presenta un decrecimiento paulatino, para luego en un segundo tramo crecer bruscamente, incluso cambiando de signo, lo que no se corresponde totalmente con el comportamiento esperado, ya que el valor del coeficiente b corresponde a un número estrictamente positivo, según las definiciones hechas en la sección 3.4. Esto puede deberse a que la red neuronal priorizó su entrenamiento en los datos más bajos, ya que en la sección 5.1 se puede observar que los datos para el porcentaje de esmectita presentan una amplia diferencia entre el primer y segundo cuartil, mientras que esta diferencia no es tan pronunciada entre el segundo y tercer cuartil. Con respecto al coeficiente b , se observa una tendencia global a decrecer, lo que se condice con los resultados esperados, ya que la presencia de arcillas (en este caso la esmectita) disminuye la velocidad de sedimentación de las muestras, haciendo que el valor de f_{bk} sea menor a medida que la concentración de esmectita aumenta, según lo reportado en la sección 3.5. Finalmente, el modelo CNN para el coeficiente A presenta una tendencia global creciente partir del 10% de esmectita; sin embargo, al igual que en el caso del porcentaje de clorita, este comportamiento se compensa con la forma decreciente que presenta el coeficiente b .

Con lo anterior, el modelo que mejor representa lo esperado para los coeficientes que componen la función de batch de Kynch es el modelo CNN, ya que a pesar de presentar una tendencia creciente para el coeficiente A , esto lo puede compensar con el comportamiento que presenta el coeficiente b .

Función batch de Kynch: Efecto moscovita

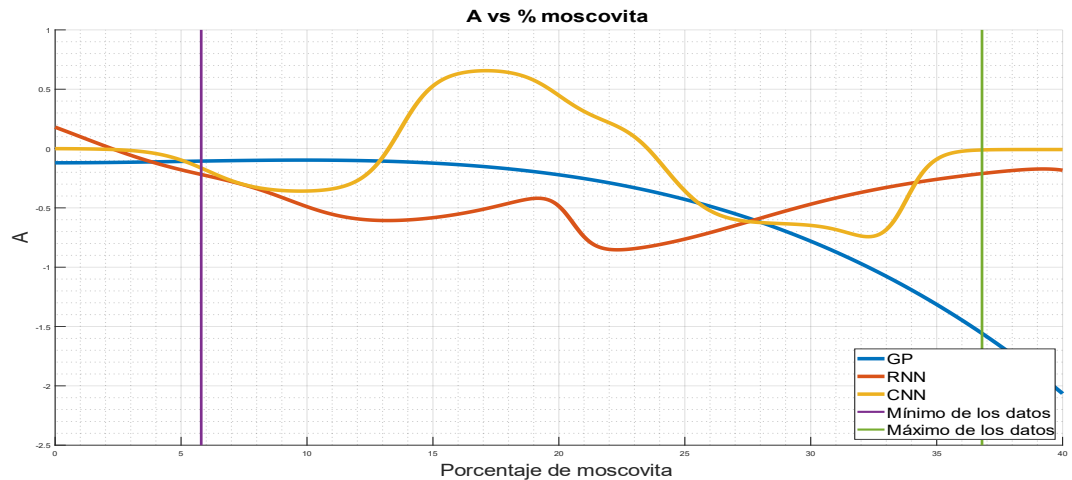


Figura 58: Simulación del coeficiente A en función del porcentaje de moscovita.

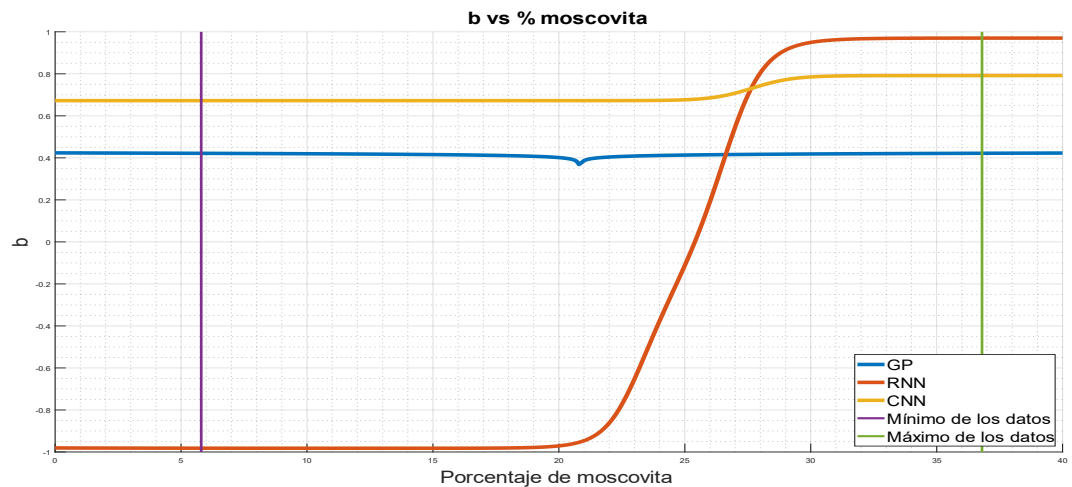


Figura 59: Simulación del coeficiente b en función del porcentaje de esmectita.

En las Figuras 58 y 59 se observa el comportamiento descrito por los modelos de Machine Learning con respecto al porcentaje de moscovita presente en la muestra para el caso de los coeficientes A y b . En cuanto al coeficiente A , los modelos RNN y CNN presentan un comportamiento errático, donde coinciden parcialmente en los extremos del rango de aplicación, pero distan especialmente en los porcentajes medios. Sin embargo, el modelo GP presenta una clara tendencia decreciente en todo el rango de aplicación, lo que se ajusta de mejor manera con el

comportamiento esperado para este factor. Con respecto al coeficiente b , se observa que tanto el modelo GP como CNN presentan un comportamiento principalmente constante, mientras que el modelo RNN presenta 2 zonas relativamente constantes, con un cambio brusco entre el 20 y 30% de moscovita. Con lo anterior, el modelo que mejor representa lo esperado para los coeficientes que componen la función de batch de Kynch es el modelo GP, considerando que posee un comportamiento definido en todo el rango de aplicación y se condice con la teoría en cuanto a su efecto en la velocidad de sedimentación de la muestra.

Función batch de Kynch: Efecto D_2

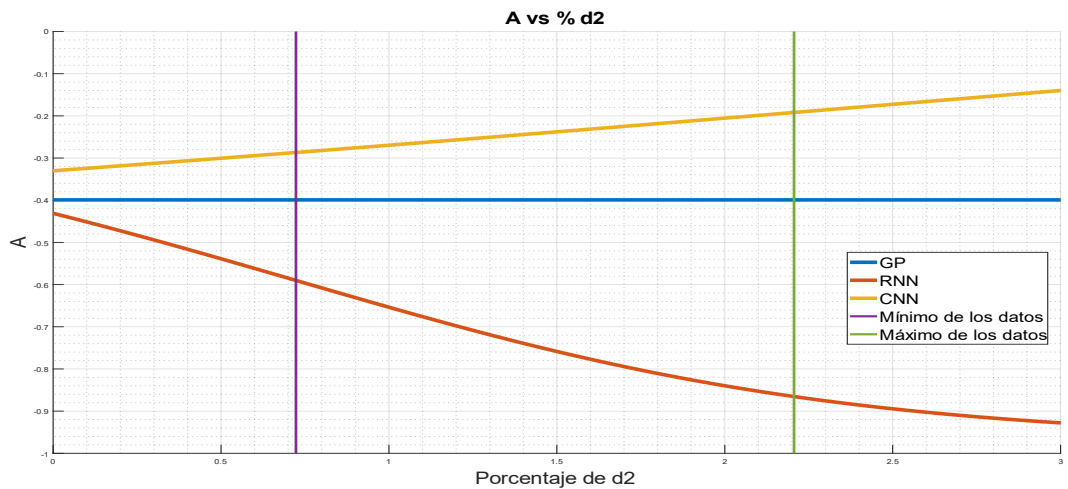


Figura 60: Simulación del coeficiente A en función de D_2 .

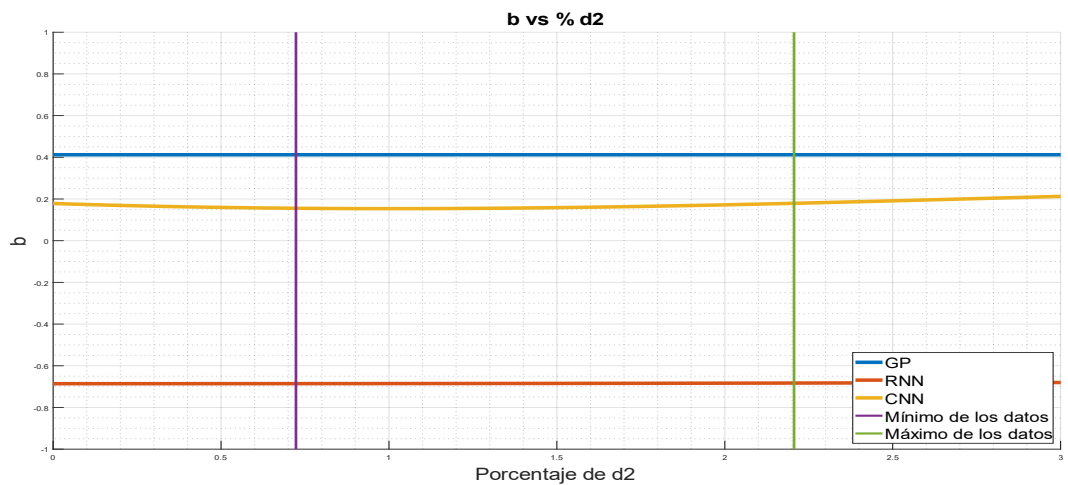


Figura 61: Simulación del coeficiente b en función de D_2 .

En las Figuras 60 y 61 se observa el comportamiento descrito por los modelos de Machine Learning con respecto a D_2 presente en la muestra para el caso de los coeficientes A y b . Para el caso del coeficiente A se observa que el modelo GP se mantiene constante, mientras que el modelo RNN decrece y el modelo CNN crece con una pendiente menor al modelo RNN. En este sentido, el modelo CNN no se comporta como es esperado, mientras que el modelo RNN muestra una disminución de este coeficiente, debido a que la velocidad de sedimentación se ve afectada si la proporción de material fino en los relaves aumenta. Por otro parte, para el coeficiente b se observa que los 3 modelos presentan un comportamiento mayoritariamente constante, por lo que en este caso todos los modelos no representan una variación del coeficiente b dada una variación en el contenido de material fino. Sin embargo, para el modelo RNN se observa un valor de b negativo, lo cual no concuerda con el comportamiento fenomenológico. Con lo anterior, no se puede definir cuál de los 3 modelos representa de mejor manera los datos simulados, ya que ninguno cumple en su totalidad con lo esperado en la teoría de sedimentación.

Función estrés efectivo del sólido: Efecto caolinita

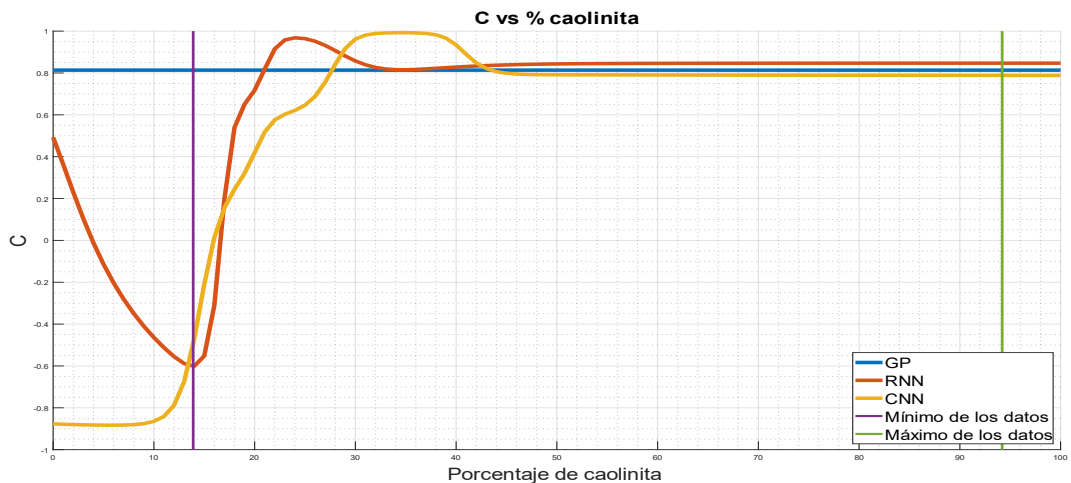


Figura 62: Simulación del coeficiente C en función del porcentaje de caolinita.

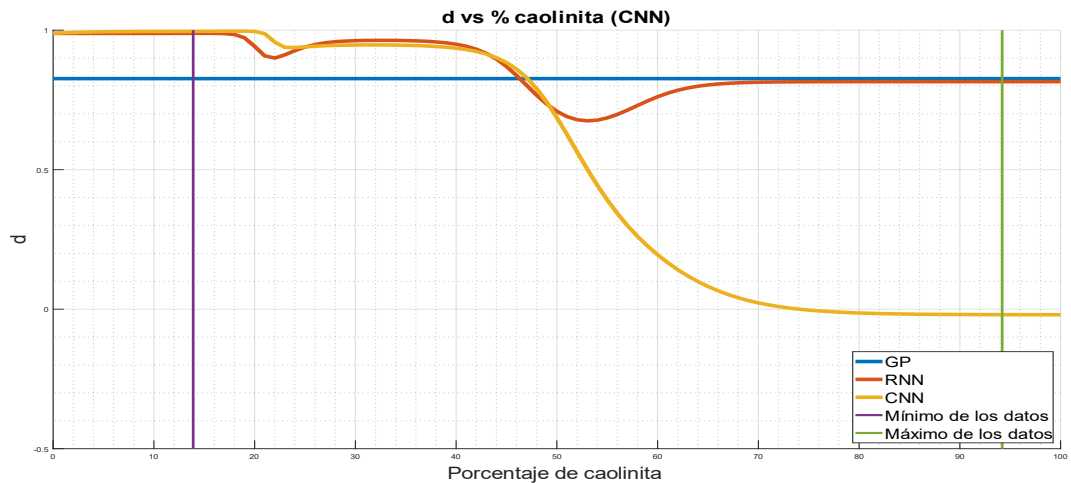


Figura 63: Simulación del coeficiente d en función del porcentaje de caolinita.

En las Figuras 62 y 63 se observa el comportamiento descrito por los modelos de Machine Learning con respecto al porcentaje de caolinita presente en la muestra para el caso de los coeficientes C y d . Para el coeficiente C , el modelo GP presenta una tendencia constante, mientras que los modelos RNN y CNN presentan un comportamiento similar, en cuanto a que comienzan con una forma creciente para luego estabilizarse aproximadamente en el mismo valor que indica el modelo GP. Esto muestra que en este caso los 3 modelos concuerdan entre sí, aunque las redes neuronales son capaces de capturar más información de los datos que el algoritmo de programación genética. Por otra parte, para el coeficiente d se hace un análisis similar, salvo que el modelo CNN no coinciden en la parte final con los demás modelos. Sin embargo, los modelos GP y RNN muestran un comportamiento similar. Con lo anterior, el modelo que mejor representa lo esperado para los coeficientes que componen la función de estrés efectivo del sólido es el modelo GP, considerando que posee un comportamiento definido en todo el rango de aplicación y se condice con la teoría en cuanto a su efecto en la velocidad de sedimentación de la muestra.

Función estrés efectivo del sólido: Efecto clorita

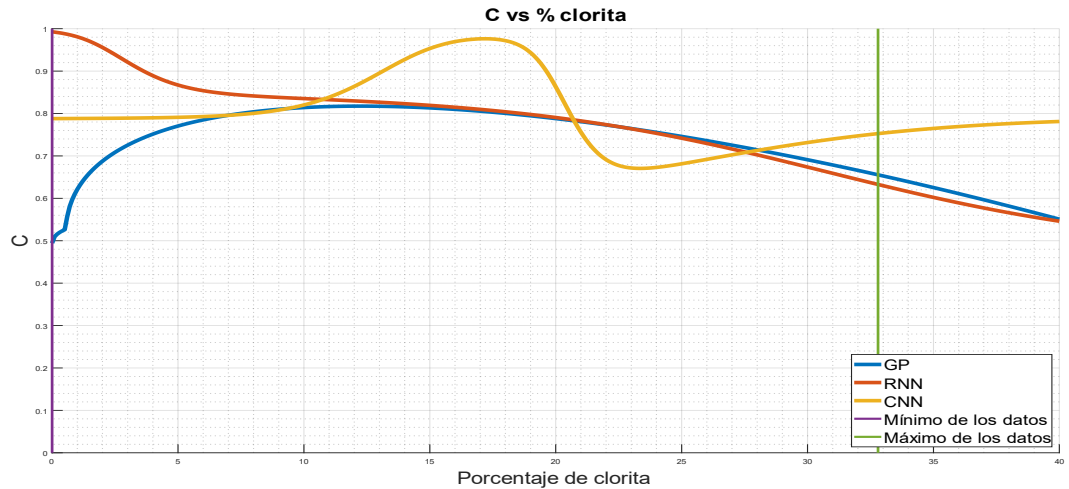


Figura 64: Simulación del coeficiente C en función del porcentaje de clorita.

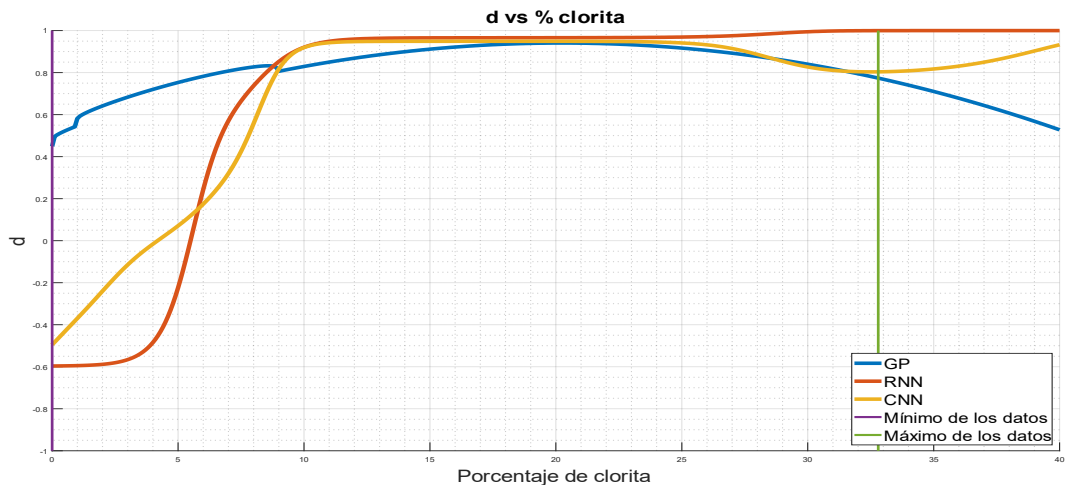


Figura 65: Simulación del coeficiente d en función del porcentaje de clorita.

En las Figuras 64 y 65 se observa el comportamiento descrito por los modelos de Machine Learning con respecto al porcentaje de clorita presente en la muestra para el caso de los coeficientes C y d . Se observa, para ambos coeficientes, una tendencia similar para los 3 modelos. Sin embargo, al comienzo de ambos gráficos es donde se puede apreciar una diferencia entre los modelos. La forma de la curva descrita por el modelo CNN es errática para ambos coeficientes, en especial para C , y cabe destacar la similitud entre los modelos GP y RNN, en particular para el coeficiente C .

Con lo anterior, el modelo que mejor representa lo esperado para los coeficientes que componen la función de estrés efectivo del sólido es el modelo GP, considerando que posee un comportamiento definido en todo el rango de aplicación y se condice con la teoría de la sedimentación.

Función estrés efectivo del sólido: Efecto esmectita

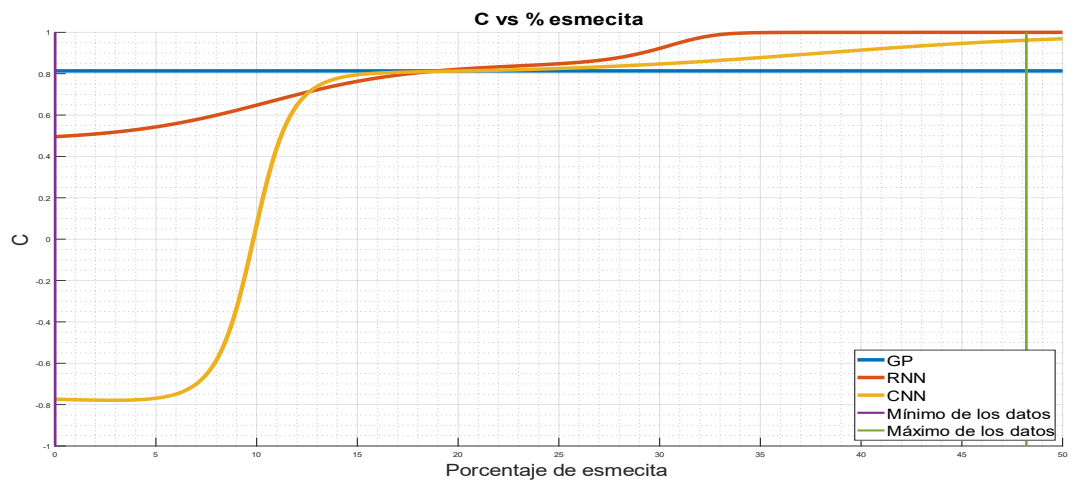


Figura 66: Simulación del coeficiente C en función del porcentaje de esmectita.

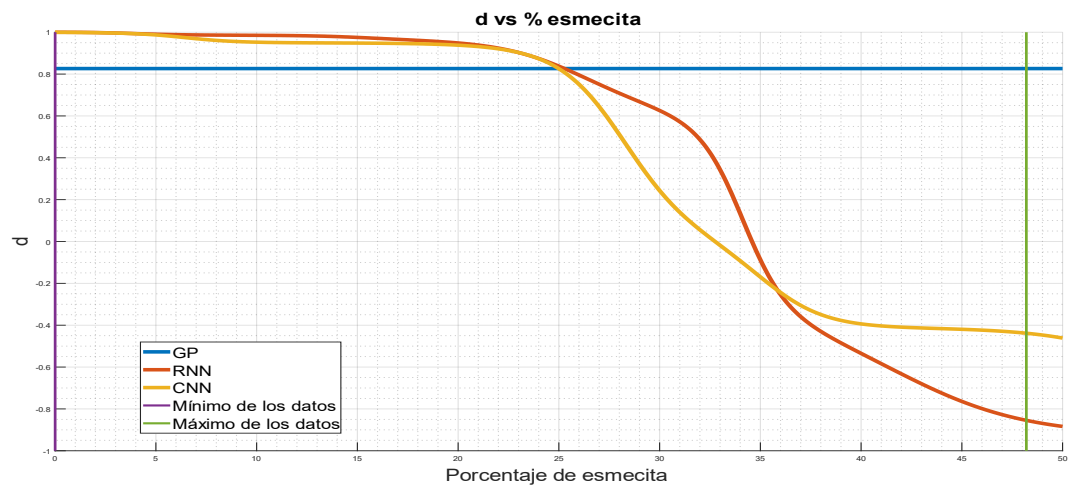


Figura 67: Simulación del coeficiente d en función del porcentaje de esmectita.

En las Figuras 66 y 67 se observa el comportamiento descrito por los modelos de Machine Learning con respecto al porcentaje de esmectita presente en la muestra para el caso de los coeficientes C y d . Se observa, para el coeficiente C , una tendencia similar para los 3 modelos. Sin embargo, al comienzo de ambos gráficos es donde se puede apreciar una diferencia más notoria entre los modelos, mostrando al modelo CNN más alejado de los demás. Se puede apreciar que el modelo RNN logra una tendencia creciente a lo largo del gráfico, donde el modelo GP muestra aproximadamente un valor promedio constante a ésta. Para el coeficiente, el modelo GP no muestra variaciones para esta variable, mientras que los modelos RNN y CNN presentan comportamientos similares, distanciándose notoriamente en 2 tramos.

Con lo anterior, el modelo que mejor representa lo esperado para los coeficientes que componen función de estrés efectivo del sólido es el modelo RNN, considerando que posee un comportamiento definido en todo el rango de aplicación y se condice con la teoría de la sedimentación.

Función estrés efectivo del sólido: Efecto moscovita

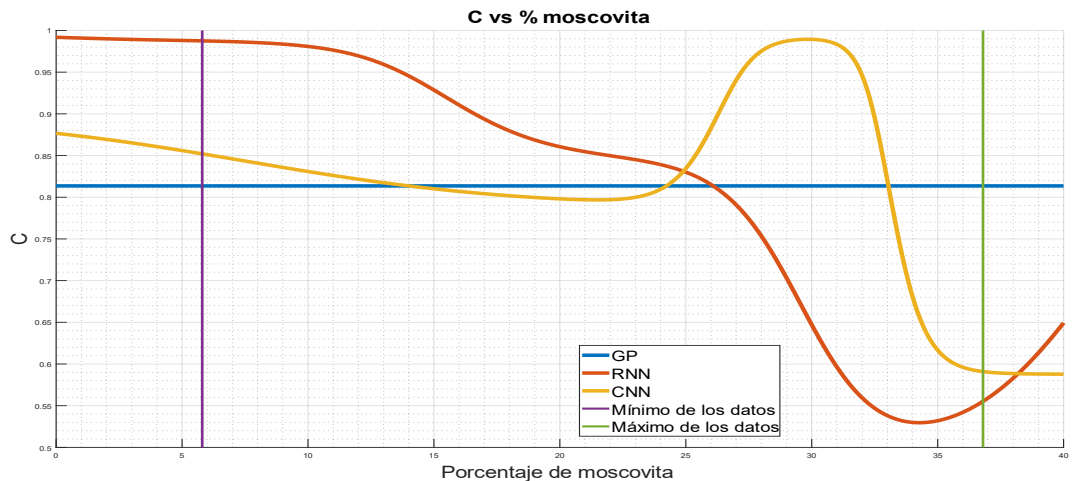


Figura 68: Simulación del coeficiente C en función del porcentaje de moscovita.

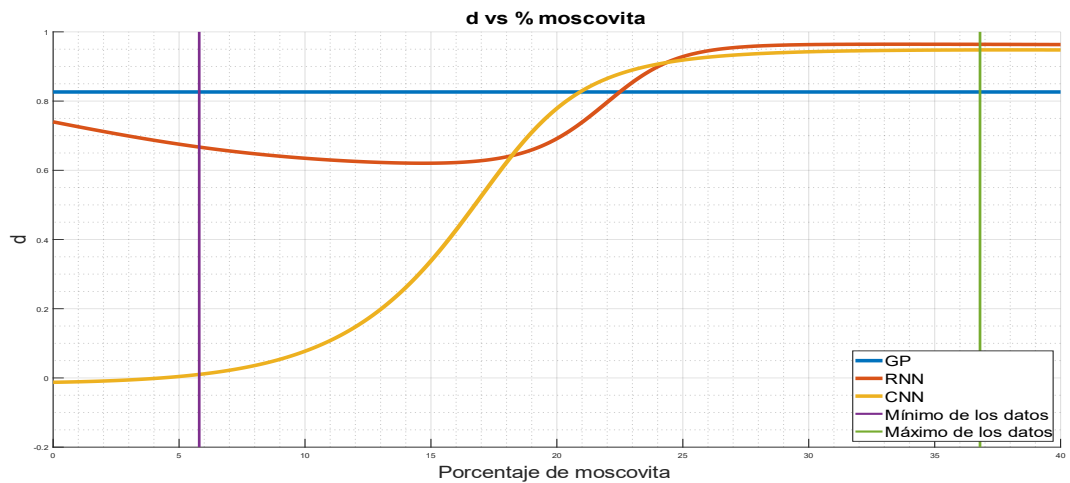


Figura 69: Simulación del coeficiente d en función del porcentaje de moscovita.

En las Figuras 68 y 69 se observa el comportamiento descrito por los modelos de Machine Learning con respecto al porcentaje de moscovita presente en la muestra para el caso de los coeficientes C y d . Para ambos coeficientes, el modelo GP no presenta variaciones al aumentar el porcentaje de moscovita en la muestra, entregando un valor constante. Con respecto al modelo RNN, se tiene que una forma notoriamente decreciente dentro del rango de los datos para el coeficiente C , mientras que para el coeficiente d la curva es creciente, con una leve disminución en la primera mitad del gráfico. Finalmente, para el modelo CNN se observa un comportamiento similar al modelo anterior, aunque en el final del gráfico de C tiene una forma que no coincide con lo esperado, mientras que para el coeficiente d se observa una tendencia similar a RNN, aunque muy alejado en especial al comienzo, distando de los demás modelos. Con lo anterior, el modelo que mejor representa lo esperado para los coeficientes que componen función de estrés efectivo del sólido es el modelo RNN, considerando que posee un comportamiento definido en todo el rango de aplicación y se condice con la teoría de la sedimentación.

Función estrés efectivo del sólido: Efecto D_2

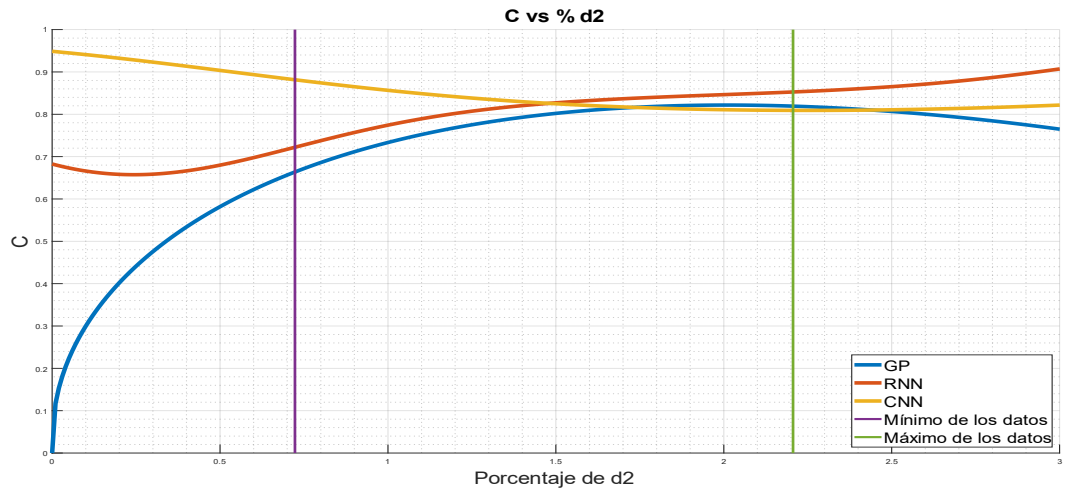


Figura 70: Simulación del coeficiente C en función de D_2 .

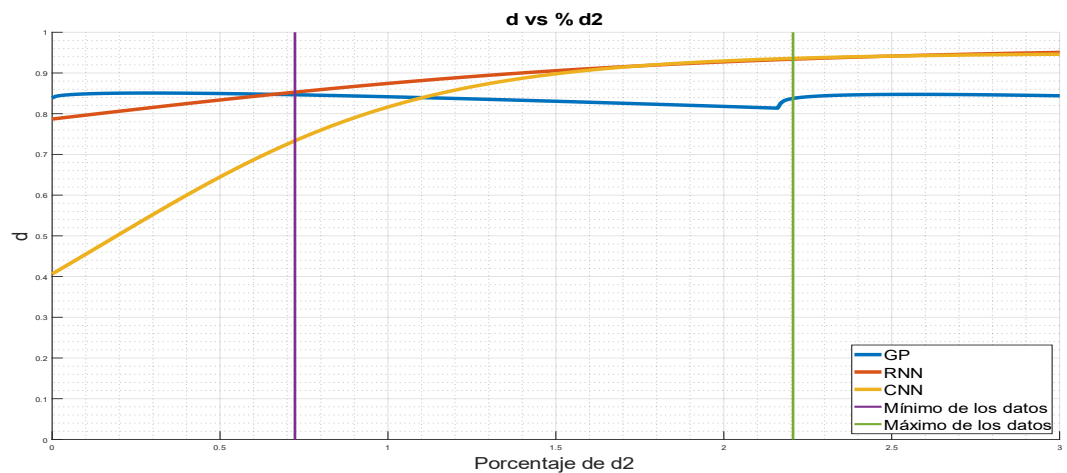


Figura 71: Simulación del coeficiente d en función de D_2 .

En las Figuras 70 y 71 se observa el comportamiento descrito por los modelos de Machine Learning con respecto a D_2 presente en la muestra para el caso de los coeficientes C y d . En ambos gráficos se puede observar que, para valores bajos de D_2 , hay mayor discrepancia entre sus valores, mientras que a medida va aumentando esta variable, los modelos coinciden en mayor medida, siendo los modelos RNN y CNN quienes presentan mejor coincidencia en este rango. Para el modelo GP se observa un comportamiento creciente para el coeficiente C , donde en el final comienza

a ser decreciente, mientras que el modelo RNN es principalmente creciente y CNN presenta una forma opuesta a GP, es decir, primero decreciente y luego creciente. Finalmente, para el coeficiente d , el modelo CNN está alejado en comparación a la diferencia entre los otros modelos, coincidiendo notoriamente con RNN a partir de $D_2 = 1,5\%$.

Con lo anterior, el modelo que mejor representa lo esperado para los coeficientes que componen función de estrés efectivo del sólido es el modelo RNN, considerando que posee un comportamiento definido en todo el rango de aplicación y se condice con la teoría de la sedimentación.

Función de dosificación: Efecto caolinita

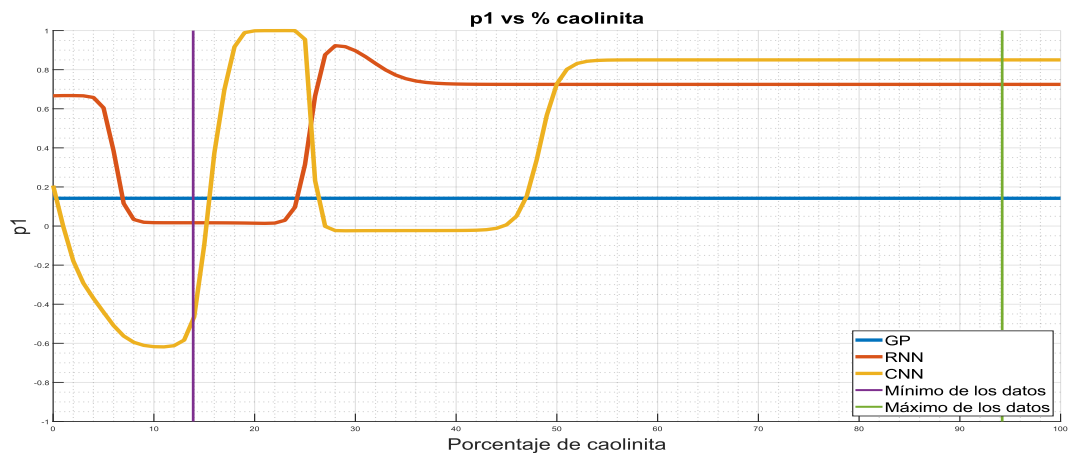


Figura 72: Simulación del coeficiente p_1 en función del porcentaje de caolinita.

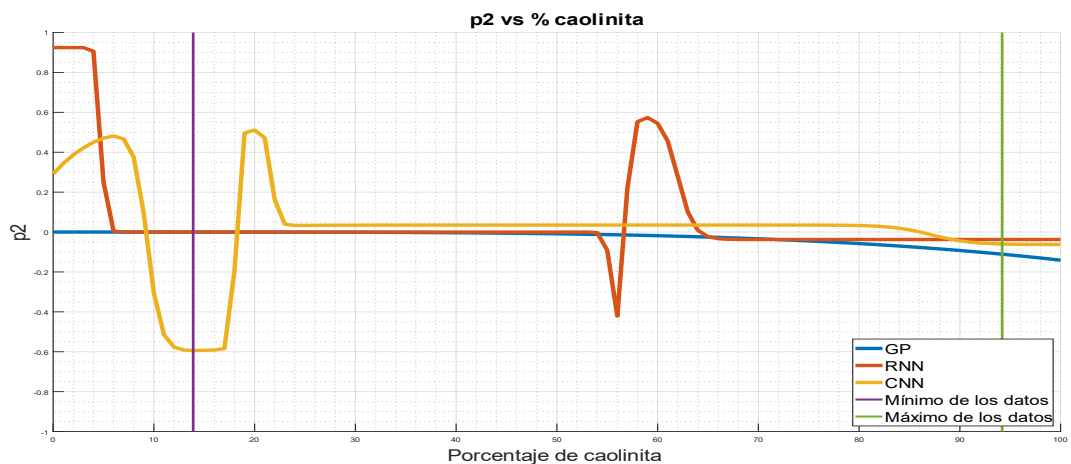


Figura 73: Simulación del coeficiente p_2 en función del porcentaje de caolinita.

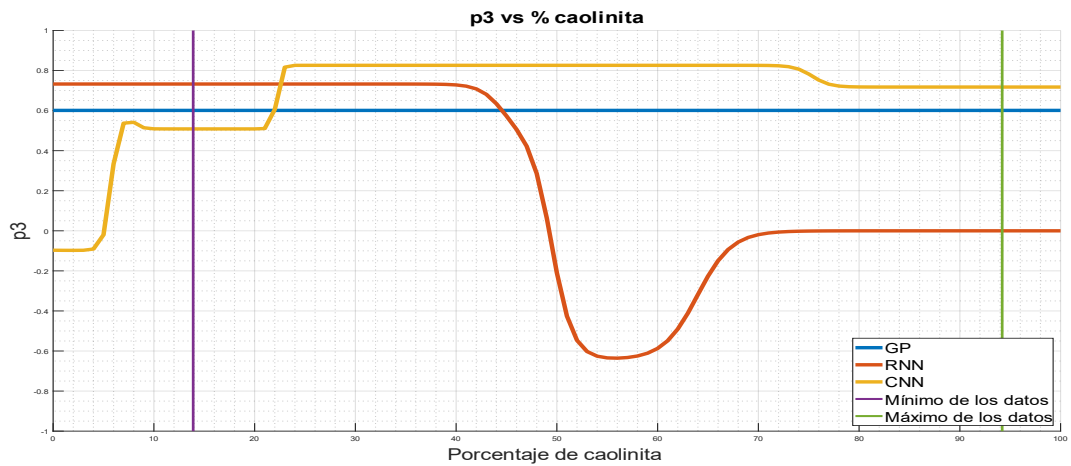


Figura 74: Simulación del coeficiente p_3 en función del porcentaje de caolinita.

En las Figuras 72, 73 y 74 se observa el comportamiento descrito por los modelos de Machine Learning con respecto al porcentaje de caolinita presente en la muestra para el caso de los coeficientes p_1, p_2 y p_3 . Se observa que para los 3 gráficos, el modelo GP presenta un valor constante, por lo que la dosificación no se ve afectada por el porcentaje de caolinita, mientras que en general, los modelos RNN y CNN presentan un comportamiento oscilatorio y errático a lo largo de los gráficos, lo que puede deberse a que los modelos fuerzan sus resultados para aproximar de mejor manera a los datos experimentales. Esto muestra que ninguno de los modelos refleja el comportamiento de los datos y no son capaces de representar los datos.

Función de dosificación: Efecto clorita

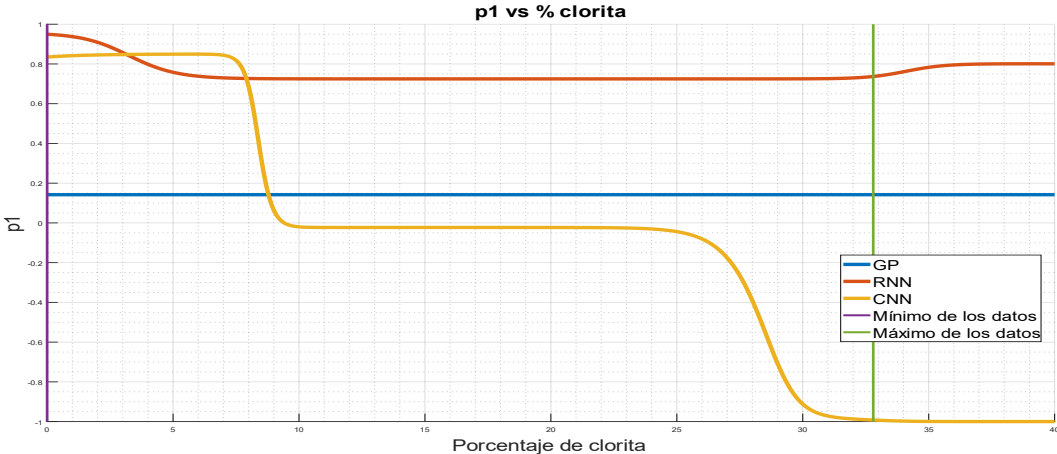


Figura 75: Simulación del coeficiente p_1 en función del porcentaje de clorita.

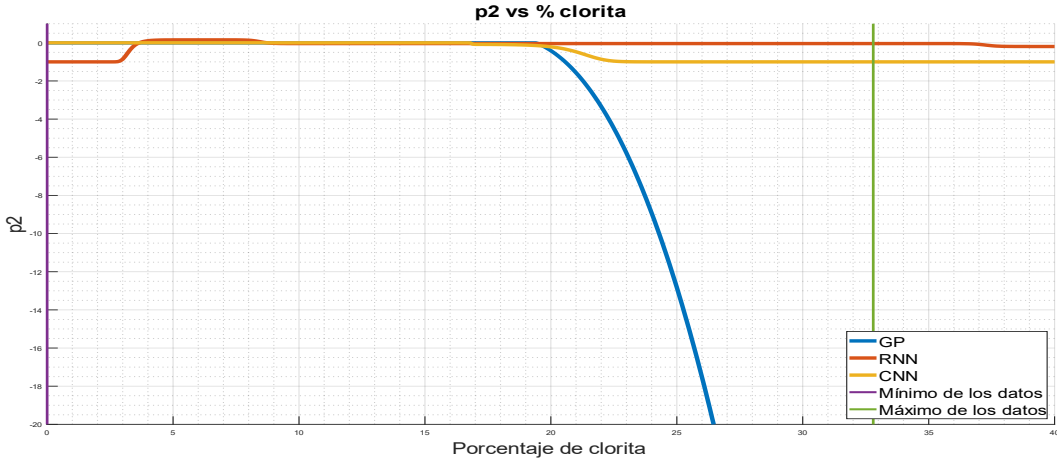


Figura 76: Simulación del coeficiente p_2 en función del porcentaje de clorita.

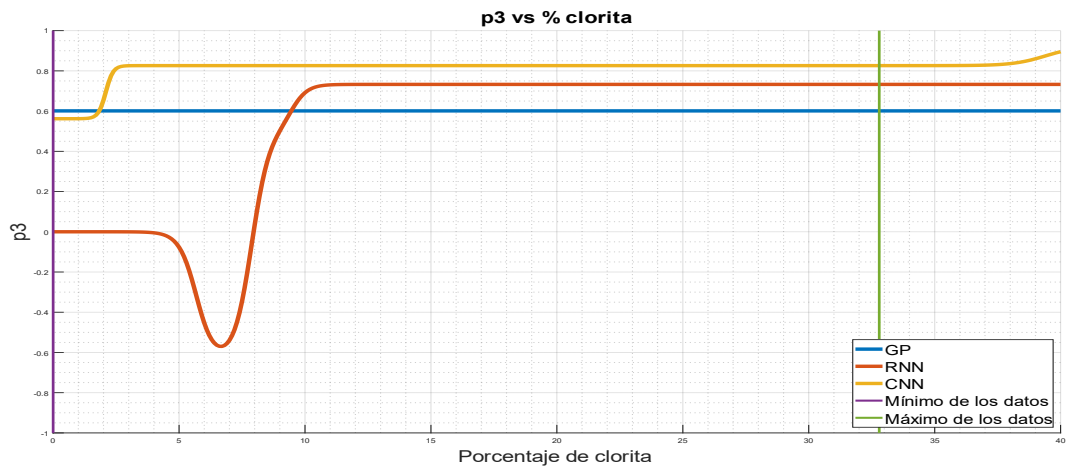


Figura 77: Simulación del coeficiente p_3 en función del porcentaje de clorita.

En las Figuras 75, 76 y 77 se observa el comportamiento descrito por los modelos de Machine Learning con respecto al porcentaje de clorita presente en la muestra para el caso de los coeficientes p_1 , p_2 y p_3 . Con respecto al coeficiente p_1 se observa que el modelo GP no presenta variaciones con respecto al porcentaje de clorita, mientras que el modelo RNN presenta una forma relativamente constante en todo el rango de aplicación, con pequeñas variaciones en sus extremos, y en cuanto al modelo CNN se observa una forma escalonada, que no se condice con ninguna de las anteriores. Para el coeficiente p_2 se observa que los modelos RNN y CNN tienen un comportamiento por tramos constante, coincidiendo mayormente entre los 4 y 20% de clorita, es decir, gran parte del rango de aplicación. En cuanto al modelo GP se ve una disminución abrupta a partir del 20%, alejándose significativamente de la predicción de los demás modelos. Finalmente, para el coeficiente p_3 , tanto GP como CNN tienen un comportamiento constante en la mayor parte del rango, en cambio el modelo RNN presenta una forma constante, luego una concavidad para establecerse en un valor cercano al de los otros modelos.

Función de dosificación: Efecto esmectita

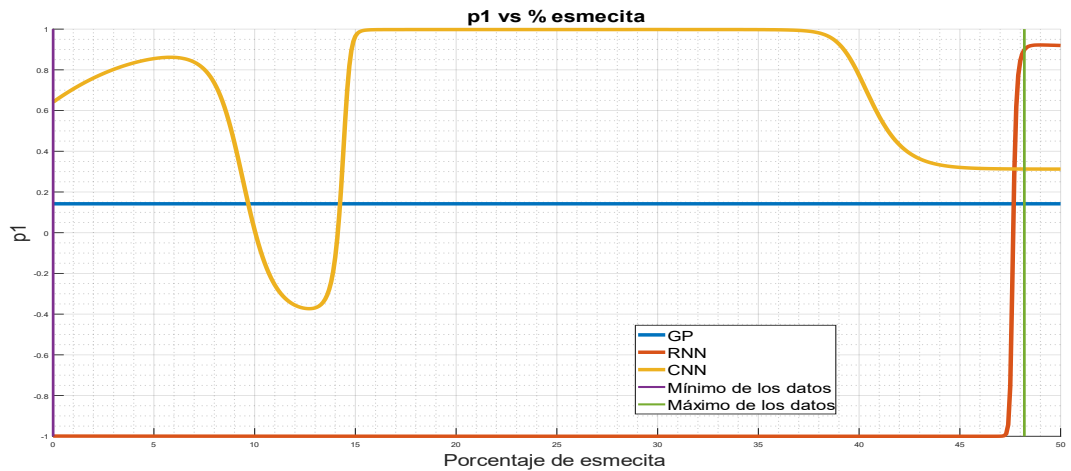


Figura 78: Simulación del coeficiente p_1 en función del porcentaje de esmectita.

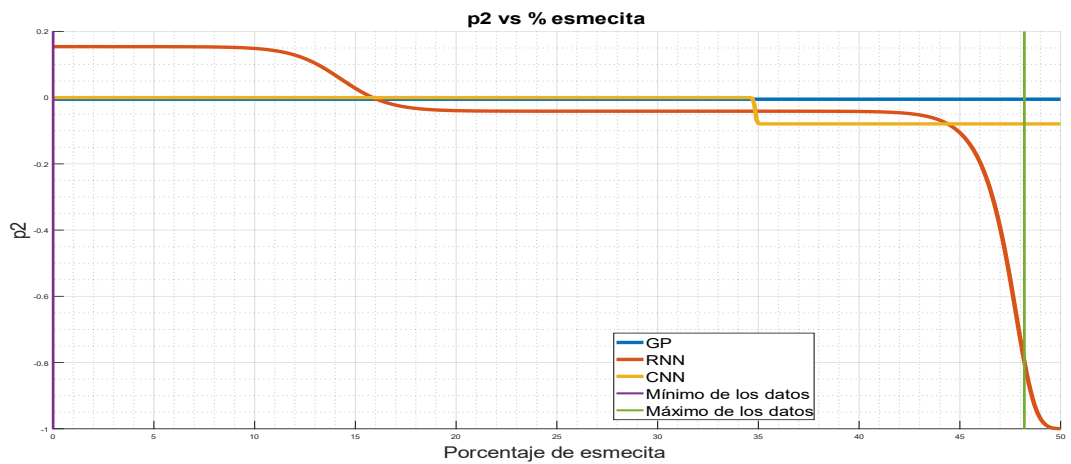


Figura 79: Simulación del coeficiente p_2 en función del porcentaje de esmectita.

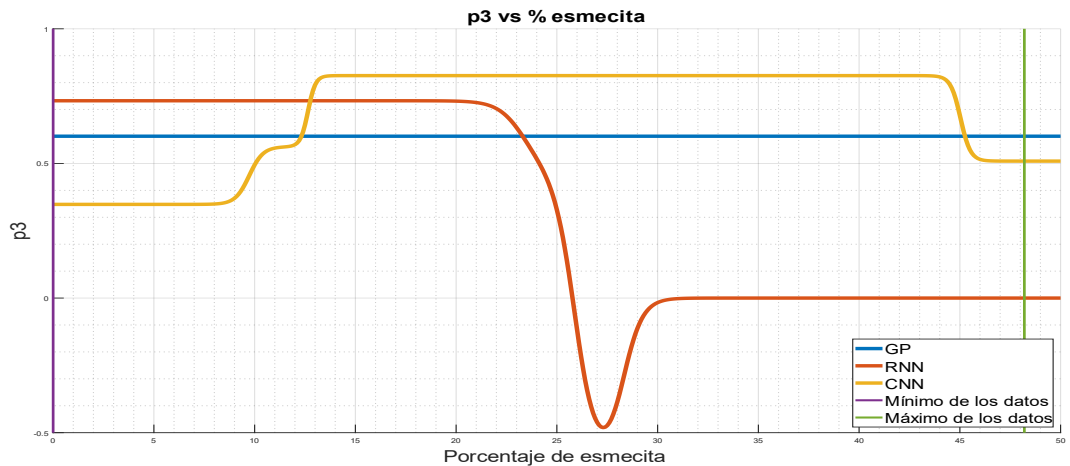


Figura 80: Simulación del coeficiente p_3 en función del porcentaje de esmecita.

En las Figuras 78, 79 y 80 se observa el comportamiento descrito por los modelos de Machine Learning con respecto al porcentaje de esmecita presente en la muestra para el caso de los coeficientes p_1 , p_2 y p_3 . Para el coeficiente p_1 , el modelo GP presenta un valor constante a lo largo del gráfico, el modelo RNN presenta un valor constante prácticamente en todo el rango, aunque en este caso el valor es negativo, por lo que la parábola construida a partir de este coeficiente cuadrático será cóncava, lo cual no corresponde al comportamiento observado, mientras que el modelo CNN presenta un comportamiento más errático, por lo que no coincide con los valores presentados por los demás modelos, oscilando en un pequeño rango el signo del valor numérico. Para el coeficiente p_2 se observa un rango aproximadamente entre el 15 y 45% de esmecita donde los 3 modelos presentan valores similares y constantes, aunque en general coinciden en cuanto a su comportamiento en todo el rango de aplicación. Finalmente, para el coeficiente p_3 , el modelo presenta un valor constante, RNN presenta una convexidad entre el 20 y 30% de esmecita, presentando dos valores constantes fuera de este rango, donde el primero es positivo y el segundo es cero.

Función de dosificación: Efecto moscovita

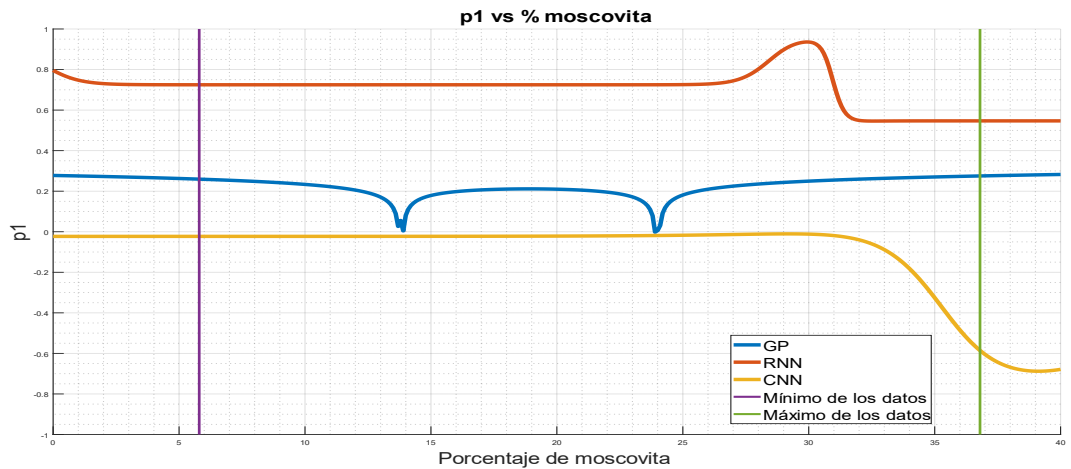


Figura 81: Simulación del coeficiente p_1 en función del porcentaje de moscovita.

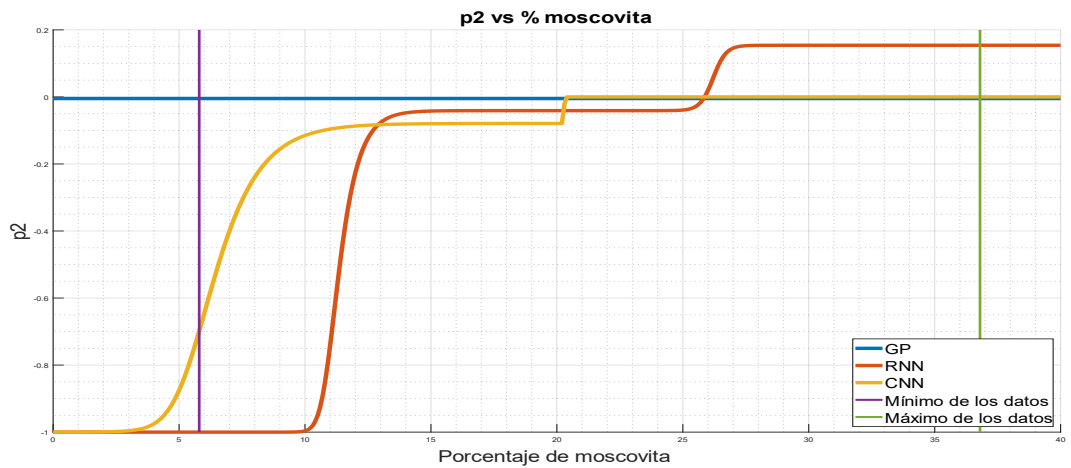


Figura 82: Simulación del coeficiente p_2 en función del porcentaje de moscovita.

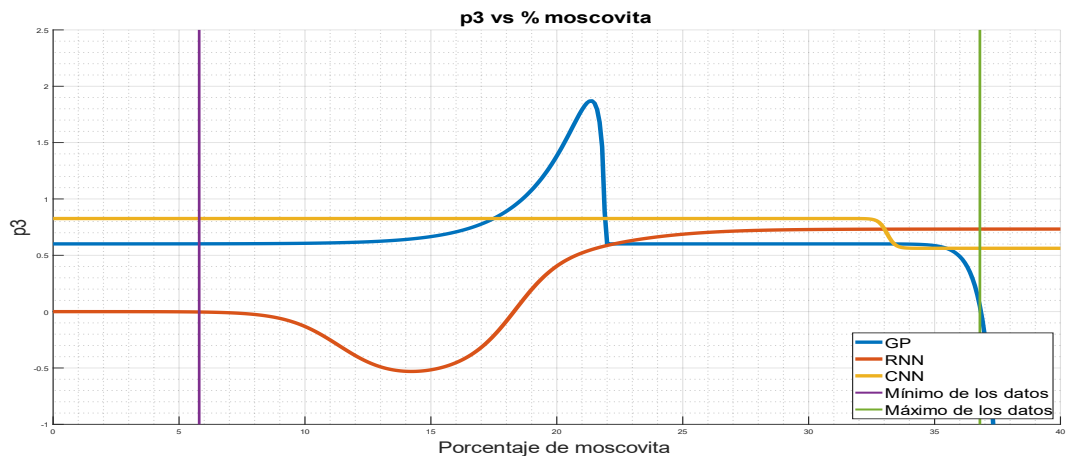


Figura 83: Simulación del coeficiente p_3 en función del porcentaje de moscovita.

En las Figuras 81, 82 y 83 se observa el comportamiento descrito por los modelos de Machine Learning con respecto al porcentaje de moscovita presente en la muestra para el caso de los coeficientes p_1 , p_2 y p_3 . Para el coeficiente p_1 se tiene un comportamiento parcialmente constante para todos los modelos, aunque para el modelo GP se observan 2 mínimo locales muy significativos (específicamente en 14% y 24% de moscovita, aproximadamente), el modelo RNN presenta valores positivos en todo el rango, con 2 escalones claros dentro del rango, y, por otro lado, el modelo CNN presenta un valor constante en gran parte del gráfico, y una disminución en el último tramo, aunque presenta valores negativos para este parámetro. Para el coeficiente p_2 se presenta un valor constante para el modelo GP, mientras que los modelos RNN y CNN tienen un comportamiento exponencial estacionario, llegando a valores similares entre los 3 modelos. Finalmente, el coeficiente p_3 , el modelo GP presenta un máximo pronunciado en 22 – 23% de moscovita aproximadamente, con valores constante antes y después de ello, el modelo RNN presenta un valor constante, seguido por una curva decreciente y luego creciente para estabilizarse en otro valor constante mayor al primero, y el modelo CNN presenta un comportamiento en escalones, mostrando un valor constante en la mayor parte del gráfico, el cual disminuye aproximadamente a los 33% de moscovita.

Función de dosificación: Efecto D_2

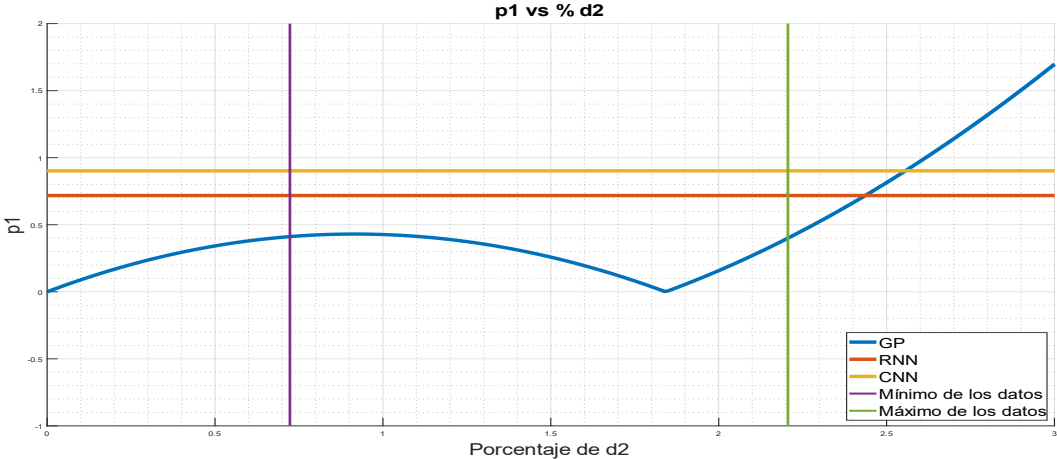


Figura 84: Simulación del coeficiente p_1 en función de D_2 .

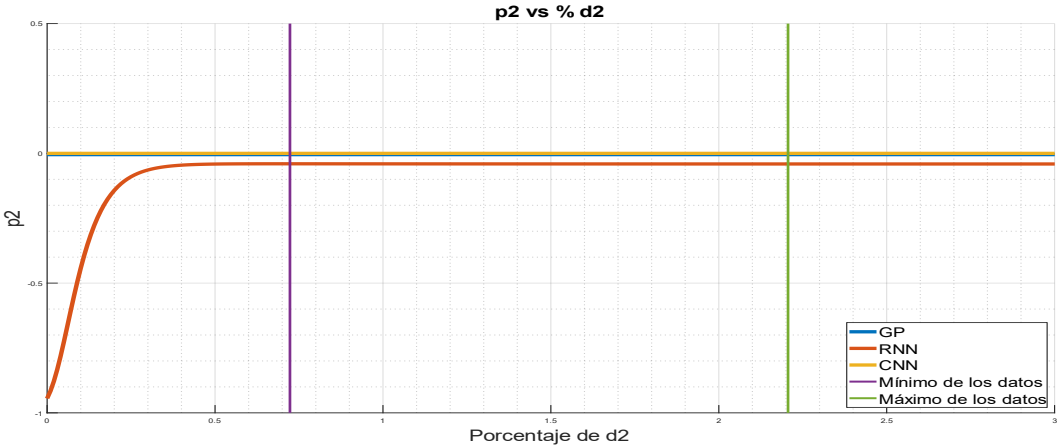


Figura 85: Simulación del coeficiente p_1 en función de D_2 .

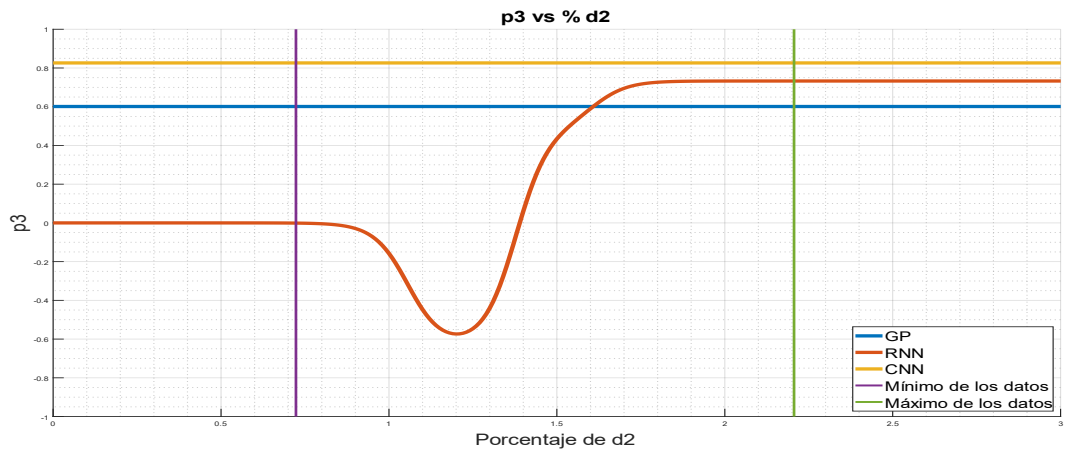


Figura 86: Simulación del coeficiente p_1 en función de D_2 .

En las Figuras 84, 85 y 86 se observa el comportamiento descrito por los modelos de Machine Learning con respecto a D_2 presente en la muestra para el caso de los coeficientes p_1, p_2 y p_3 . En el coeficiente p_1 , los modelos RNN y CNN presentan una forma constante, mientras que el comportamiento del modelo GP es errático para este caso. Para el coeficiente p_2 , los modelos GP y CNN coinciden tanto en el comportamiento constante como en el valor exhibido, mientras que el modelo RNN presenta una forma creciente, estabilizándose en un valor menor pero similar al de los otros modelos. Por último, para el coeficiente p_3 los modelos GP y CNN presentan un valor constante, aunque con valores distintos, mientras que el modelo RNN presenta un valor constante en un principio, seguido con un decrecimiento y luego un crecimiento hasta un valor que se encuentra entre los valores presentados por los otros modelos.

Concentración crítica ϕ_c : Efecto caolinita

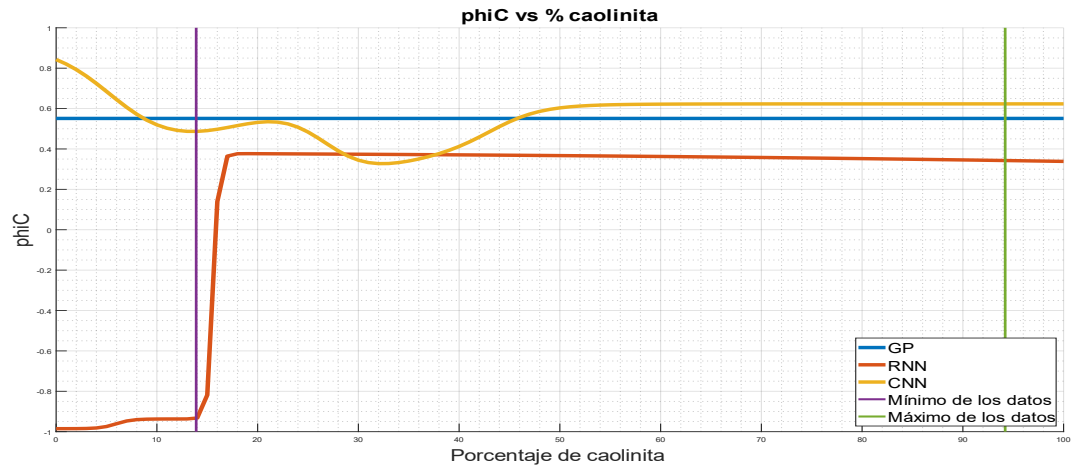


Figura 87: Simulación de la concentración crítica ϕ_c en función del porcentaje de caolinita.

En la Figura 87 se observa el comportamiento descrito por los modelos de Machine Learning con respecto al porcentaje de caolinita presente en la muestra para el caso de los coeficientes ϕ_c . Se observa que el modelo GP no presenta variaciones al aumentar el porcentaje de caolinita, por lo que su valor se mantiene constante. Para el modelo RNN se observa un aumento abrupto al comienzo del rango de los datos, llegando a un valor constante por prácticamente todo el rango. Finalmente, para el modelo CNN se presenta un comportamiento oscilatorio para luego estabilizarse a partir de la mitad del rango. Con esto se puede notar que esta variable principalmente no presenta variaciones para los modelos.

Con lo anterior, el modelo que mejor representa lo esperado para la concentración crítica es el modelo GP, considerando que posee un comportamiento definido en todo el rango de aplicación y se condice con la teoría de la sedimentación.

Concentración crítica ϕ_c : Efecto clorita

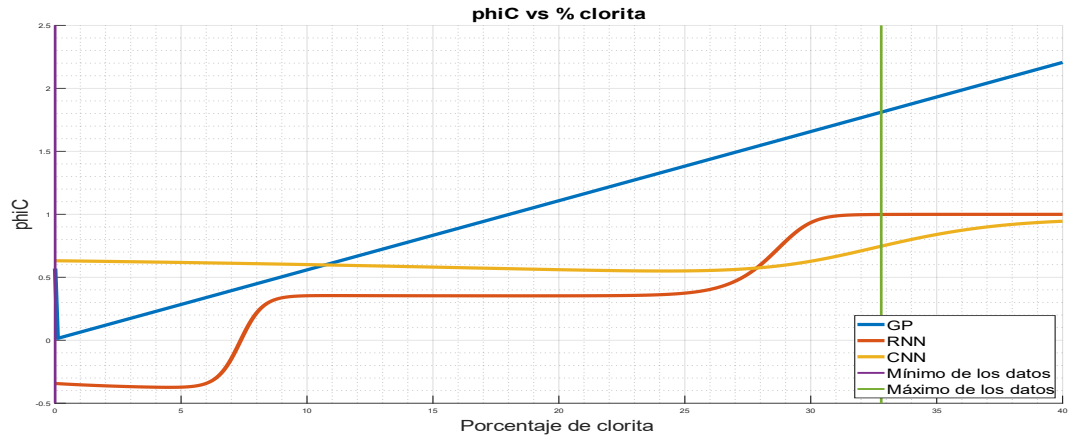


Figura 88: Simulación de la concentración crítica ϕ_c en función del porcentaje de clorita.

En la Figura 88 se observa el comportamiento descrito por los modelos de Machine Learning con respecto al porcentaje de clorita presente en la muestra para el caso de los coeficientes ϕ_c . Se observa que el modelo GP presenta una forma lineal creciente, es decir, a medida que aumenta el porcentaje de clorita, aumenta el valor de la concentración crítica, por lo que la muestra sedimenta más lento. El modelo RNN presenta un comportamiento creciente, aunque por escalones, mientras que el modelo CNN presenta un valor constante en la mayor parte del gráfico.

Con lo anterior, el modelo que mejor representa lo esperado para la concentración crítica es el modelo RNN, considerando que posee un comportamiento definido en todo el rango de aplicación y se condice con la teoría de la sedimentación.

Concentración crítica ϕ_c : Efecto esmectita

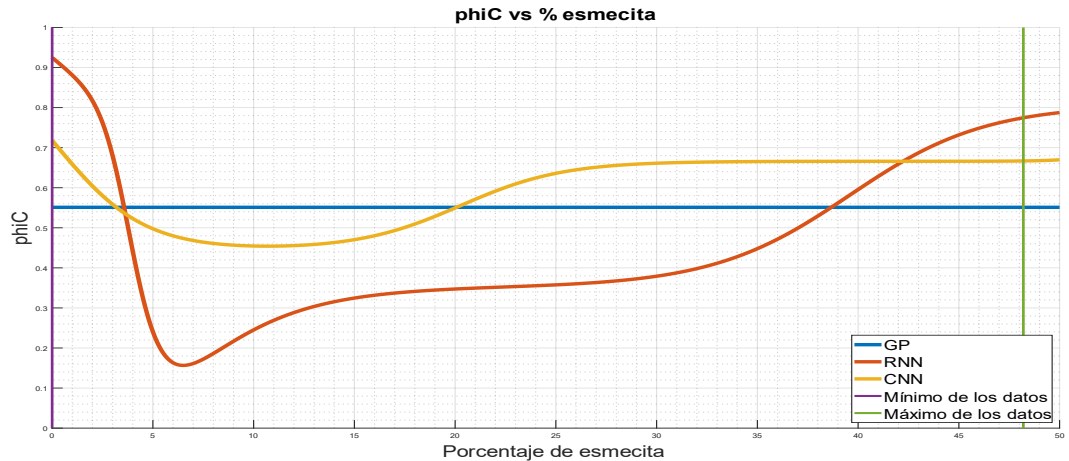


Figura 89: Simulación de la concentración crítica ϕ_c en función del porcentaje de esmectita.

En la Figura 89 se observa el comportamiento descrito por los modelos de Machine Learning con respecto al porcentaje de esmectita presente en la muestra para el caso de los coeficientes ϕ_c . El modelo GP presenta un valor constante, por lo que el porcentaje de esmectita no influye sobre la concentración crítica, para este modelo. El modelo RNN presenta un comportamiento oscilatorio, pero principalmente tiene una tendencia creciente. Finalmente, el modelo CNN presenta una forma cóncava y luego se estabiliza en un valor mayor que GP.

Con lo anterior, el modelo que mejor representa lo esperado para la concentración crítica es el modelo GP, considerando que posee un comportamiento definido en todo el rango de aplicación y se condice con la teoría de la sedimentación.

Concentración crítica ϕ_c : Efecto moscovita

En la Figura 90 se observa el comportamiento descrito por los modelos de Machine Learning con respecto al porcentaje de moscovita presente en la muestra para el caso de los coeficientes ϕ_c . El modelo GP no presenta variaciones a medida que aumenta el porcentaje de moscovita, mostrando un valor constante. Para el modelo RNN se muestra un comportamiento

decreciente por escalones al igual que para el modelo CNN, aunque este último desciende un escalón más, llegando a un valor negativo para la concentración crítica, lo que es físicamente incorrecto.

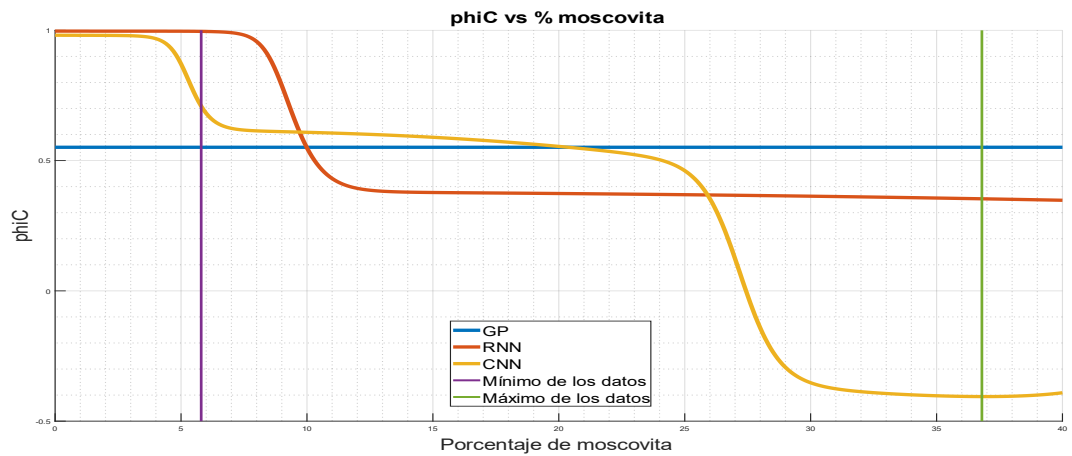


Figura 90: Simulación de la concentración crítica ϕ_c en función del porcentaje de moscovita.

Con lo anterior, el modelo que mejor representa lo esperado para la concentración crítica es el modelo RNN, considerando que posee un comportamiento definido en todo el rango de aplicación y se condice con la teoría de la sedimentación.

Concentración crítica ϕ_c : Efecto D_2

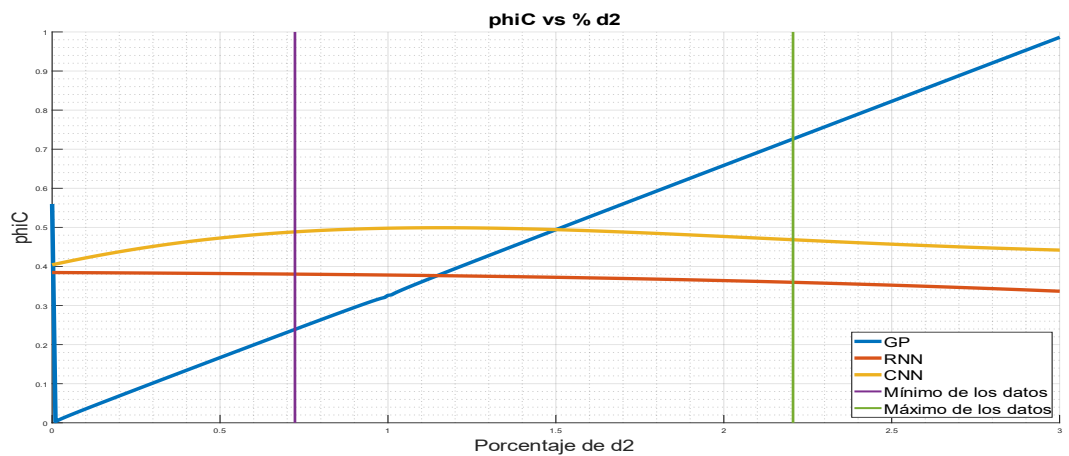


Figura 91: Simulación de la concentración crítica ϕ_c en función de D_2 .

En la Figura 91 se observa el comportamiento descrito por los modelos de Machine Learning con respecto a D_2 presente en la muestra para el caso de los coeficientes ϕ_c . Se observa que el modelo GP presenta una forma lineal creciente, es decir, a medida que aumenta el porcentaje de clorita, aumenta el valor de la concentración crítica, por lo que la muestra sedimenta más lento. El modelo RNN presenta un comportamiento decreciente, mientras que el modelo CNN presenta un crecimiento al principio del gráfico, mostrando luego un decrecimiento progresivo en la mayor parte del gráfico.

Con lo anterior, el modelo que mejor representa lo esperado para la concentración crítica es el modelo RNN, considerando que posee un comportamiento definido en todo el rango de aplicación y se condice con la teoría de la sedimentación.

5.7 Comentarios Generales

El desempeño y capacidad de generalización de cada modelo se debe evaluar desde distintos enfoques. Por un lado, con respecto a los tiempos de cómputo se debe distinguir en 2 etapas, las cuales se muestran sus resultados en la Tabla 19.

Tabla 19: Tiempos de cómputo de los modelos diseñados.

Modelos	Entrenamiento [<i>min</i>]	Ejecución [<i>s</i>]
GP	205	3,5
RNN	290	4,0
CNN	300	4,1

El tiempo de entrenamiento se refiere al tiempo necesario para poder desarrollar cada modelo. En el caso de los algoritmos de programación genética, estos fueron diseñado con 50 generaciones y 500 individuos teniendo una duración promedio de 3 horas y 25 minutos, mientras que las redes neuronales, diseñadas con 2 capas (capa 1 con 10 neuronas y capa 2 con 1 neurona), este proceso duró en promedio 4 horas y 40 minutos para las redes neuronales recurrentes, y 4 horas y 50 minutos para las redes

neuronales convolucionales. Cabe destacar que estos valores corresponden a promedios de los algoritmos diseñados, ya que cada variable modelada de las funciones constitutivas demoró tal tiempo, es decir, para poder simular todas variables mediante programación genética el proceso de entrenamiento demora aproximadamente 27 horas, mediante redes neuronales recurrentes demora aproximadamente 39 horas, y mediante redes neuronales convolucionales demora aproximadamente 40 horas. Se puede observar una gran diferencia entre la programación genética con respecto a los demás modelos, teniendo un tiempo al menos 12 horas menor al de los otros modelos. Por otro lado, el tiempo de ejecución corresponde al tiempo que demora una simulación utilizando los modelos ya entrenados. Se puede observar que entre los 3 modelos no existe una gran diferencia, siendo para términos prácticos indiferente la elección de cualquiera de ellos.

Con respecto al estudio de los parámetros estadísticos presentado en la sección 5.4, se puede notar un mejor desempeño para las redes neuronales con respecto al algoritmo de programación genética, siendo éste claramente presentado en la mayor parte de los casos, presentando valores más cercanos a 1 para el coeficiente de correlación R^2 y menores órdenes de magnitud para el caso del error cuadrático medio $RMSE$. Esto puede deberse a que para todas los parámetros de las funciones constitutivas, los modelos creados a partir de programación genética solo consideran a 2 de las 5 variables de entradas disponibles para modelar los datos experimentales. Este hecho está relacionado con la implementación del método de Jackknife ya que este disminuye el sobreajuste que podrían potencialmente presentar los modelos si considerasen más variables para su diseño. Con esto, el algoritmo podría buscar combinaciones que presenten un mejor desempeño en cuanto a efectividad, sin embargo, esto conllevaría a un entrenamiento con más individuos y generaciones, lo que según el párrafo anterior implicaría un mayor tiempo de entrenamiento y ejecución.

Finalmente, para el análisis de tendencias de la sección 5.5, los modelos en general presentan una buena capacidad para representar las tendencias esperadas de crecimiento y decrecimiento de las variables con respecto a cada una de las variables. Sin embargo, para la mayoría de las muestras, los modelos tienen problemas para simular correctamente a los valores bajos de sus respectivas series de datos, entregando valores lejanos a los que

realmente corresponden. Esta es una de las características que se repite en la gran parte de modelos, independiente del algoritmo utilizado para su diseño, pero principalmente se puede observar en los provenientes de programación genética y de hecho es una de las razones por las que el desempeño de estos modelos es menor al de las redes neuronales artificiales. Analizando este apartado en relación a las redes neuronales, se observa que las redes recurrentes presentan una mayor capacidad para representar las tendencias de los datos en general, sin embargo, las redes convolucionales muestran un comportamiento esperado con respecto a su capacidad para hallar patrones, distinguiendo y representando de mejor manera a los datos más bajos de las series, pudiendo así reconocer distintos subgrupos dentro de las muestras estudiadas. En el Anexo C se puede observar el comportamiento de las funciones constitutivas para cada una de las muestras, comparando los valores experimentales con los simulados por cada modelo.

6. Conclusiones

El comportamiento de los relaves en cuanto a sus funciones constitutivas depende de diversos factores, de los cuales, según la información reportada en literatura y la experiencia por parte de la empresa, algunos de los más importantes corresponden a los porcentajes de arcillas, es decir, los componentes de filosilicatos con un tamaño muy fino. El trabajo realizado se desarrolló a partir de los datos experimentales de muestras de relaves a partir de ensayos de sedimentación, por lo que el desempeño de éste puede ser mejorado al añadir más muestras a las series de datos, y/o más datos de las mismas muestras, lo que permitiría reducir los errores experimentales que puedan haberse presentado.

El objetivo de este trabajo consistió en desarrollar modelos a partir de algoritmos de Machine Learning, en este caso, se utilizaron 2 tipos de redes neuronales artificiales (recurrentes y convolucionales) y el algoritmo de programación genética, Si bien el estudio de estos modelos permite explicar los fenómenos más relevantes del proceso, todavía quedan condiciones en las que el diseño puede mejorar, como por ejemplo, porcentajes muy bajos de arcillas y material fino. Es por esto que la elección de estos modelos sirve como una aproximación al fenómeno total.

Con respecto al estudio de variabilidad y dispersión de las series de datos se pudo determinar ciertas consideraciones para el modelamiento de p_1 y p_2 , debido a que la distribución de datos perjudicaba el correcto funcionamiento de los algoritmos de Machine Learning y los métodos de remuestreo. Estas series efectivamente presentan mayores discrepancias entre lo que establece cada modelo matemático con respecto a los datos experimentales, lo cual su desempeño se vería beneficiado agregando ensayos que permitan dar cuenta de una distribución adecuada. Para el caso de las demás variables simuladas, los modelos no muestran mayor problema a la hora de predecir los datos, teniendo las redes neuronales una mayor efectividad en este ámbito, especialmente la red recurrente, quien presenta un general menores diferencias entre lo simulado y lo real.

En la etapa de entrenamiento de los modelos basados en Machine Learning, para el caso de las redes neuronales, se lograron diseñar 2 tipos de redes neuronales. En primer lugar, se encuentran los modelos basados en redes neuronales recurrentes, los cuales presentan desempeños aceptables para sistemas que se vean beneficiados de una retroalimentación, y en segundo lugar, también se utilizaron redes neuronales convolucionales, que se aplican para el reconocimiento de patrones dentro de series de datos. Ambas herramientas se diseñaron de manera similar, con los mismos parámetros de diseño, utilizando como método de selección de variables el método Lasso en validación cruzada para reducir evitar el sobreajuste por el exceso de parámetros en las redes neuronales, obteniendo 8 modelos basados en cada tipo de red neuronal.

Por otro lado, para la implementación del algoritmo de programación genética, se utilizaron parámetros que significaron un costo computacional similar al de las redes neuronales, en cuanto a recursos usados y tiempo de cómputo, además de ciertas restricciones que se respaldan en la fenomenología del sistema. Se utiliza el método jackknife para evitar el sobreajuste y como método de penalización se implementa el método de mínimos cuadrados penalizados. En esta etapa se obtuvieron 8 expresiones matemáticas que dan cuenta de la modelación a través de programación genética de los coeficientes de interés de las funciones constitutivas del sistema.

La validación de datos se realizó en función de la efectividad de cada modelo con respecto a los datos experimentales disponibles. En general, los modelos diseñados a partir de redes neuronales, tanto recurrentes como convolucionales, presentan valores más cercanos a los esperados, teniendo para cada caso buenos resultados con respecto a la predicción. Por otro lado, los modelos basados en programación genética presentan en general un desempeño aceptable, considerando para todos los casos a 2 de las 5 variables de entrada para el diseño de los modelos.

Los criterios estadísticos indican que, a nivel de desempeño global de los modelos, las redes neuronales recurrentes muestran mejores valores, tanto en el coeficiente de correlación R^2 y el error cuadrático medio $RMSE$. Sin embargo, las redes convolucionales presentan rendimientos similares a las

anteriores, en particular a los resultados obtenidos del coeficiente de correlación. Con respecto a los modelos GP se tiene que sus coeficientes no son tan buenos como los obtenidos de redes neuronales, aunque en la mayoría de los casos sí presentan desempeños aceptables para predecir los datos y caracterizar las tendencias generales en cuanto a las distintas variables de entrada. Se presentan dificultades por parte de todos los modelos para predecir los valores relacionados a la función de dosificación (p_1, p_2, p_3), lo cual fue un punto a considerar desde el estudio de la distribución de los datos, lo cual permitió detectar de buena manera esta complicación.

Sobre el análisis fenomenológico y matemático (en el caso de la programación genética), los modelos son capaces de representar las tendencias y comportamientos esperados con respecto a los datos experimentales, destacando para este caso los modelos basados en redes neuronales convolucionales, donde se puede observar una mejor predicción en los rangos más bajos de las series de datos, es decir, para porcentajes bajos de arcillas y material fino. En la mayoría de los casos, los algoritmos basados en programación genética presentan valores constantes debido a la elección de variables a utilizar para el diseño de los modelos.

Por otra parte, en cuanto a los tiempos relacionados al entrenamiento de los modelos, las redes neuronales mostraron tiempo de diseño mayores a los de los algoritmos de programación genética, aunque en temas prácticos no corresponde a diferencias muy significativas, ya que corresponde a una etapa que debe realizar sólo una vez, y en el caso de querer actualizar la base de datos experimentales para añadir más información a los modelos ya creados, esta etapa es de carácter periódico. Con respecto a los tiempos de ejecución, no se presentan diferencias notorias en cuanto a los modelos, siendo indiferente bajo este apartado escoger un algoritmo por sobre otro.

Por esto, se debe considerar que la elección de los modelos depende de varios factores, que dan cuenta del desempeño global de dichos modelos. Sin embargo, en relación a lo descrito anteriormente, los modelos basados en redes neuronales recurrentes exhiben características favorables en cuanto a nivel de predicción, tiempo de entrenamiento de ejecución, y análisis fenomenológico, correspondiendo de esta manera a los modelos con mayor

desempeño y capacidad de generalización, con los datos disponibles al momento de entrenar los algoritmos.

Finalmente, se debe destacar que los resultados y conclusiones de este trabajo pueden ser aplicables a distintos problemas de investigación, como ejemplo, la metodología utilizada para el entrenamiento y diseño de los modelos podría ser utilizada en escenarios similares, tanto en la industria minera como en otros escenarios que se requiera una aplicación similar. Por otro lado, los resultados del desempeño y capacidad de generalización de los modelos estudiados entregan valiosa información acerca del comportamiento de las muestras de relaves. Tanto las metodologías utilizadas podrían extenderse al estudio de otras muestras presentes en las faenas mineras.

Bibliografía

- [1] Zhang, H., Hoang N., Xuan-Nam, B., Trung, N., "Developing a novel artificial intelligence model to estimate the capital cost of mining projects using deep neural network-based ant colony optimization algorithm," *J. Resources Policy*, vol. 66, 2020.
- [2] Yamakawa, S., "The Application of Artificial Intelligence to Electricity Power System," *IFAC Proceedings Volumes*, vol. 46, 1990.
- [3] Chetty, G., "Smart Phone Based Data Mining for Human Activity Recognition," *Procedia Computer Science*, vol.46, 2015.
- [4] Martinson, R., "Relaves de alto espesamiento: Desafíos en el diseño del sistema de transporte," 2015. [Online]. Available: <https://www.mch.cl/reportajes/relaves-de-alto-espesamiento-desafios-en-el-diseno-del-sistema-de-transporte/#>. [Accessed 18 mayo 2020].
- [5] Fundación Tecnológica, "Espesamiento de pulpas," *Boletín Minero*, 2016.
- [6] A. Mining, "How to take advantage of artificial intelligence in the mining industry," 2019. [Online]. Available: <https://www.australianmining.com.au/features/how-to-take-advantage-of-artificial-intelligence-in-the-mining-industry/>. [Accessed 18 mayo 2020].
- [7] Cameron, R., Robert, J., "Clinical applications of machine learning in the diagnosis, classification, and prediction of heart failure," *American Heart Journal*, vol. 55, 2020.
- [8] Lalmuanawma, s. Hussain, J., "Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review," *Chaos, Solitons Fractals*, vol. 139, 2020.
- [9] Minería Chilena, "Entre 2% y 4% mejoraría la eficiencia minera con Inteligencia Artificial," 2017. [Online]. Available: <https://www.mch.cl/2017/08/08/2-4-mejoraria-la-eficiencia-minera-inteligencia-artificial/>. [Accessed 18 mayo 2020].
- [10] N. Fuenzalida, "Estudio y evaluación del sistema de espesadores de relave de la división andina mediante herramientas de simulación computacional," Universidad de Chile, 2017.
- [11] Contreras, A., "La 4ta Revolución Industrial: Cómo las Empresas Mineras utilizan la Inteligencia Artificial, el Aprendizaje Automático y los Robots.," 2018. [Online]. Available:

<https://www.cochilco.cl/Listado%20Temtico/proyeccion%20agua%20mineria%20del%20cobre%202018-2029%20-%20vf.pdf>. [Accessed 19 mayo 2020].

- [12] Burger; Bustos; Concha, "Settling velocities of particulate systems: 9. Phenomenological theory of sedimentation processes: Numerical simulation of the transient behaviour of flocculated suspensions in an ideal batch or continuous thickener," *J. Miner. Process*, vol. 55, 1999.
- [13] Comisión Chilena del Cobre, 2017. [Online]. Available: <https://www.mch.cl/2017/08/08/2-4-mejoraria-la-eficiencia-minera-inteligencia-artificial/>. [Accessed 18 mayo 2020].
- [14] Rahimi; Abdollahzadeh; Rezai, "International Journal of Mineral Processing Dynamic simulation of tailing thickener at the Tabas coal washing plant using the phenomenological model," *J. Miner. Process*, vol. 154, 2016.
- [15] Portal Minero Ediciones, *Manual General de Minería y Metalurgia*. Primera Edición, 2006.
- [16] SERNAGEOMIN, *Guía técnica de operación y control de depósitos de relaves*, 2007, p. 38.
- [17] S. Partner and B. Wills, *Mineral Processing Technology an Introduction to the Practical aspects*, 2016.
- [18] P. Avaria, "División Andina: Innovación y solución para la minería," 2014. [Online]. Available: <http://www.construccionminera.cl/division-andina-innovacion-y-solucion-para-la-mineria/>. [Accessed 14 enero 2021].
- [19] Arcadis Geotecnica, "Declaración de impacto ambiental Proyecto optimización a 170 KTPD," 2009. [Online]. Available: http://bibliotecadigital.ciren.cl/bitstream/handle/123456789/6969/MMA-HUM2_0019.pdf?sequence=1&isAllowed=y. [Accessed 29 mayo 2021].
- [20] L. Valdivia and L. Vaccari, "Determinación de los parámetros óptimos para la operación del filtro tipo VPA 2040-54 en la obtención de la humedad deseada y la implementación de sistemas de gestión integrados," *Tesis para optar al Título Profesional de ingeniero químico*, Universidad Nacional San Agustín, 2014.
- [21] P. Ramachandran and A. Herle, *Solid-liquid separation*, *Chem. Eng. World*, vol. 31, n°10, pp. 75-79, 1996.
- [22] D.A. Flotation, *Chapter 3: Sludge thickening*, 2015.
- [23] M. Tanguay, P. Fawell and S. Adkins, *Modelling the impact of two different flocculants on the performance of a thickener feedwell*, *Appl. Math. Model.* vol. 38, n° 17-18, pp. 4262-4276, 2014.

- [24] Metalliferous Mining, Thickening Resource book, J. Infect. Dis. vol. 209, 2014.
- [25] FLSmidth, "Mantenimiento de espesadores," 2016.
- [26] B. Wills and J. Finch, Dewatering, Miner Process Technology. pp 417-438, 2016.
- [27] Tan; Setiawan; Bao; Bickert, "Studies on parameter estimation and model predictive control of paste thickeners," J. Process Control, vol. 28, 2015.
- [28] R. Becker and F. Concha, "Análisis del diseño y simulación de espesadores," 1982. [Online]. Available: https://www.artigos.entmme.org/download/1982/volume_2/225%20-%20Fernando%20Concha%20y%20Raúl%20Becker%20-%20Análisis%20del%20Diseño%20y%20Simulación%20de%20Espesadores.pdf. [Accessed 01 junio 2021].
- [29] G. Guzmán, "Diseño de Estrteguas de Control para Espesadores," 2009. [Online]. Available: http://repositorio.udec.cl/bitstream/11594/853/1/Tesis_Diseño_de_Estrategia_de_Control.Image.Marked.pdf. [Accessed 01 junio 2021].
- [30] C. E. Ortega Moraga, "Concentración de arcillas iónicas desde relaves por flotación inversa.," Universidad Técnica Federico Santa María, Valparaíso, 2018.
- [31] M. Hea, "Slurry rheology in wet ultrafine grinding if industrial minerals: a review," Powder Technology, 2004.
- [32] N. Arismendi, "Identificación de los elementos de gangas que modifican las propiedades reológicas y sedimentológicas de los relaves complejos," Universidad Técnica Federico Santa María, Valparaíso, 2018.
- [33] Franco-Sepúlveda, "State of the art about metaheuristics and artificial neural networks applied to open pit mining," Resources Policy. vol 60, 2019.
- [34] Varanon, "Artificial intelligence for monitoring and supervisory control of process systems," Engineering Applications of Artificial Intelligence. vol 20, 2007.
- [35] Rojas, "Inteligencia Artificial y su incorporación en minería," 2020. [Online]. Available: <https://www.antaramining.cl/noticias.html?26-inteligencia-artificial-y-su-incorporacion-en-mineria>. [Accessed 24 junio 2020].
- [36] Soft Computing and Intelligent Information Systems, "Tema 13: Programación Genética," 2014.
- [37] R. Nuzzo, "The Box Plots Alternative for visualizing quantitative data," American Academy of Physical Medicine and Rehabilitation, 2016.
- [38] D.-C. Li, W.-T. Huang, C.-C. Chen and C.-J. Chang, "Employing box plots to build high-dimensional manufacturing models for new products in TFT-LCD

- plants," *Neurocomputing Journal*, 2014.
- [39] *Soft Computing and Intelligent Information Systems*, "Tema 6: Algoritmos Genéticos I: Conceptos básicos," 2014.
- [40] De los Santos, "Tipos de aprendizaje en Machine Learning: supervisado y no supervisado," 2017. [Online]. Available: <https://empresas.blogthinkbig.com/que-algoritmo-elegir-en-ml-aprendizaje/>. [Accessed 29 mayo 2020].
- [41] Páez, "Conceptos básicos de programación genética," *Revista de la Facultad de Medicina, Universidad Nacional de Colombia*, vol. 49, 2001.
- [42] *Soft Computing and Intelligent Information Systems*, "Árboles Binarios," 2014. [Online]. Available: http://decsai.ugr.es/~jfv/ed1/tedi/cdrom/docs/arb_bin.htm. [Accessed 3 junio 2020].
- [43] IBM Knowledge Center, "El modelo de redes neuronales," 2019. [Online]. Available: <http://www.diegocalvo.es/definicion-de-red-neuronal/>. [Accessed 4 junio 2020].
- [44] Universidad de Salamanca, "Redes Neuronales," 2019. [Online]. Available: <http://avellano.fis.usal.es/~lalonso/RNA/index.htm>. [Accessed 4 junio 2020].
- [45] Calvo, "Definición de red neuronal artificial," 2017. [Online]. Available: https://www.ibm.com/support/knowledgecenter/es/SS3RA7_sub/modeler_mainhe lp_client_ddita/components/neuralnet/neuralnet_model.html. [Accessed 3 junio 2020].
- [46] Sancho, "Redes Neuronales: una visión superficial," 2019. [Online]. Available: <http://www.cs.us.es/~fsancho/?e=72>. [Accessed 4 junio 2020].
- [47] Bonet, "Redes neuronales recurrentes para el análisis de secuencias," *Revista Cubana de Ciencias Informáticas, Universidad de las Ciencias Informáticas de Cuba*, vol. 1, 2007.
- [48] Torres, "Redes neuronales recurrentes," 2019. [Online]. Available: <https://torres.ai/redes-neuronales-recurrentes/>. [Accessed 4 junio 2020].
- [49] Calvo, "Redes neuronales convolucionales," 2019. [Online]. Available: <https://torres.ai/redes-neuronales-convolucional/>. [Accessed 4 junio 2020].
- [50] L. Braga, L. Carvajal, L. Valencia and S. Carvajal, "Introducción a la Minería de datos," *E-papers*.
- [51] B. Efron, *The Jackknife, the Bootstrap and Other Resampling Plans*, 1982, p. 103.
- [52] M. Arahál, M. Soria and F. Díaz, *Técnicas de predicción con aplicaciones en ingeniería, Universidad de Sevilla*, 2006, p. 340.
- [53] D. Olson and D. Delen, *Advanced Data Mining Techniques*, Springer Science &

Business Media, 2008, p. 192.

- [54] N. Salkind, Encyclopedia of Research Design, Volume 1, SAGE Publications, 2010, p. 1719.
- [55] S. Ibisate, "Diseño de un estimulador de transferencia de carga de una furgoneta 4T/8,5T utilizando redes neuronales.," 2015. [Online]. Available: <https://core.ac.uk/download/44311092.pdf>. [Accessed 02 junio 2021].
- [56] Dindarloo, "Prediction of blast-induced ground vibrations via genetic programming," International Journal of Mining Science and Techonology. vol 25, 2015.
- [57] Ebrahimi, "Prediction and optimization of back-break and rock fragmentation using an artificial neural network and a bee colony algorithm," Bulletin of Engineering Geology and the Environment, vol 75, 2016.
- [58] Karimi, "Prediction of hydrocyclone performance using artificial neural networks," Journal of Advances in Management Research, vol. 6, 2010.
- [59] Siami-Irdemoosa, "Prediction of fuel consumption of mining dump trucks: a neural networks approach," Computers Geosciences, vol. 95, 2015.
- [60] Yu, "A hybrid GA-TS algorithm for open vehicle routing optimization of coal mines material," Expert Systems with Applications. vol 38, 2011.
- [61] D.A Flotation.
- [62] Australian Department of Industry tourism and resources, Tailings management, 2014, pp. 39-44.
- [63] P. Garrido, R. Burger and F. Concha, "Settling velocities of particulate systems: 11. Comparision of the phenomenological sedimentation-consolidation model with published experimental results," Int. J. Miner. Process. vol. 60, n° 3-4, pp. 213-227, 2000.
- [64] R. Burger, S. Evje, K. Karlsen and K. Lie, "Numerical methods for the simulation of the settling of flocculated suspensions," Chem. Eng. J. vol. 80, n° 1-3, pp.91-104, 2000.
- [65] Facultad de Estudios Superiores Cuautitlán, "Histograma," Universdiad Nacional Autónoma de México, 2014.
- [66] R. Sánchez Pedraza, "Análisis de los datos mediante herramientras gráficas," Revista Facultad de Medicina, 2000.

Anexos

Anexo A: Funciones de activación para el diseño de redes neuronales artificiales

En la Tabla 20 se muestran las funciones de activación que usualmente se utilizan para el diseño de redes neuronales con sus respectivos dominios de aplicación.

Tabla 20: Funciones comunes de activación de neuronas.

Nombre	Ecuación	Rango
Identidad	$f(x) = x$	$(-\infty, \infty)$
Paso binario	$f(x) = \begin{cases} 0 & \text{para } x < 0 \\ 1 & \text{para } x > 0 \end{cases}$	$\{0,1\}$
Logístico	$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$	$[0,1]$
Tangente hiperbólico	$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$(-1,1)$
Arcotangente	$f(x) = \tan^{-1}(x)$	$\left(\frac{-\pi}{2}, \frac{\pi}{2}\right)$
Soft Clipping	$f(\alpha, x) = \frac{1}{\alpha} \log\left(\frac{1 + e^{\alpha x}}{1 + e^{\alpha(x-1)}}\right)$	$(0,1)$
Sinc	$f(x) = \begin{cases} 1 & \text{para } x = 0 \\ \frac{\sin(x)}{x} & \text{para } x \neq 0 \end{cases}$	$[\approx -0,217234,1]$
Gaussiana	$f(x) = e^{-x^2}$	$(0,1]$

Anexo B: Resultados detallados algoritmo de programación genética

A.1 Función batch de Kynch

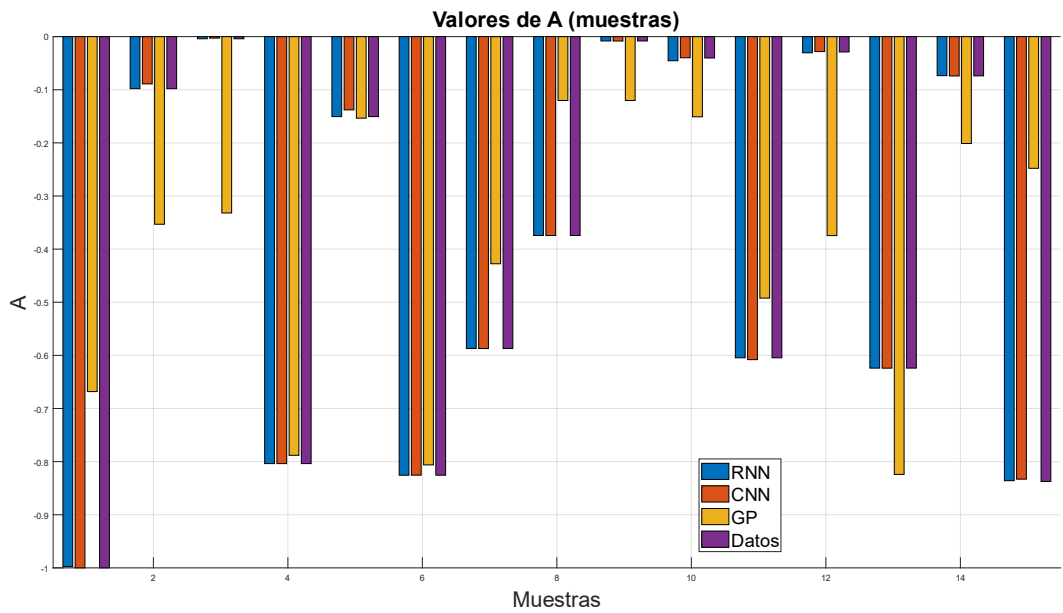


Figura 92: Valores obtenidos para cada modelo. Pre-exponencial de la función batch de Kynch.

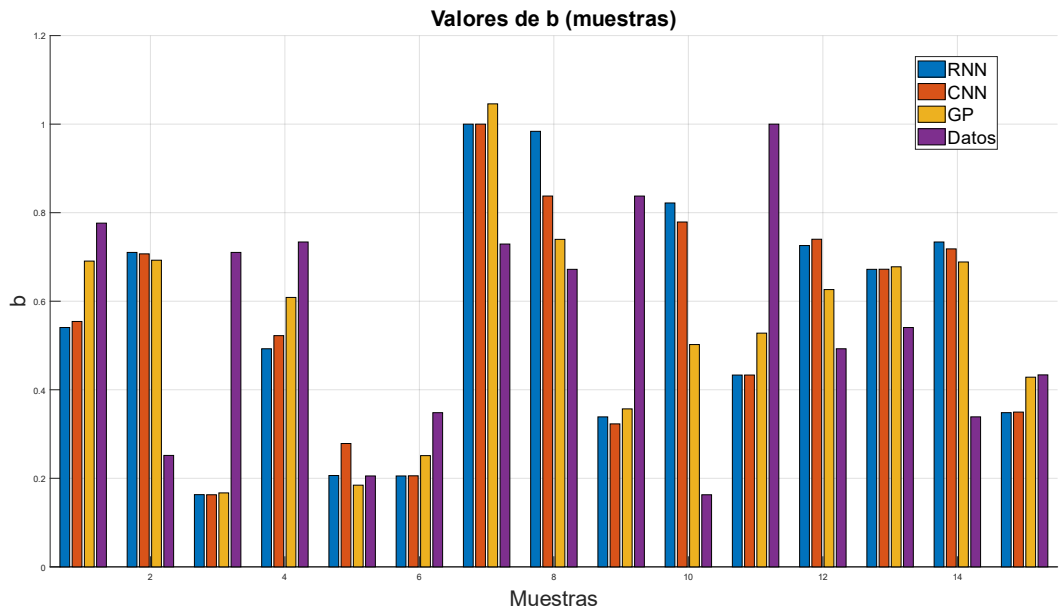


Figura 93: Valores obtenidos para cada modelo. Exponente de la función batch de Kynch.

A.2 Función estrés efectivo de sólido

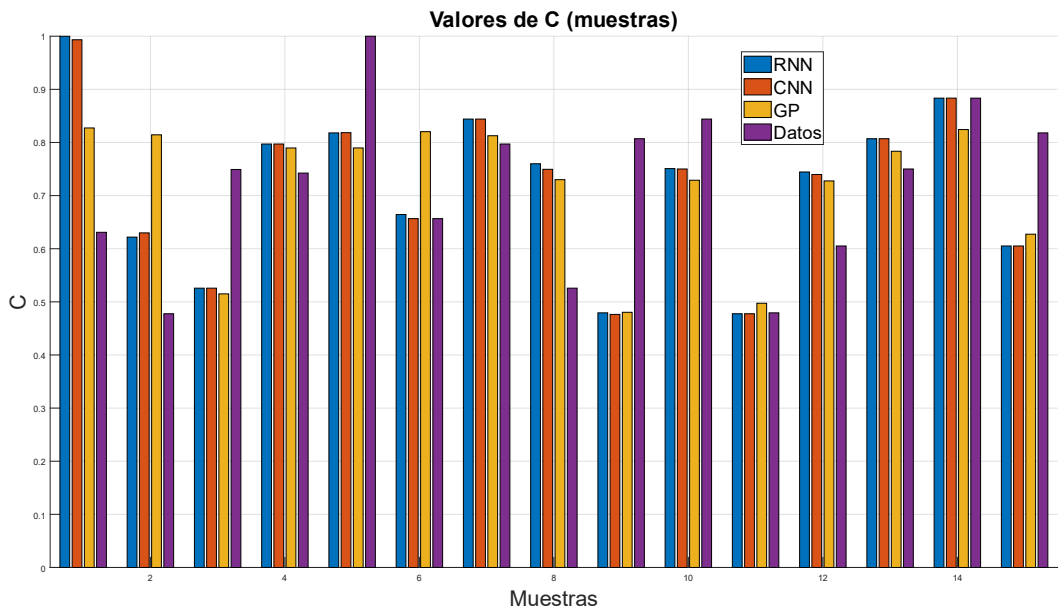


Figura 94: Valores obtenidos para cada modelo. Pre-exponencial de la función estrés efectivo del sólido.

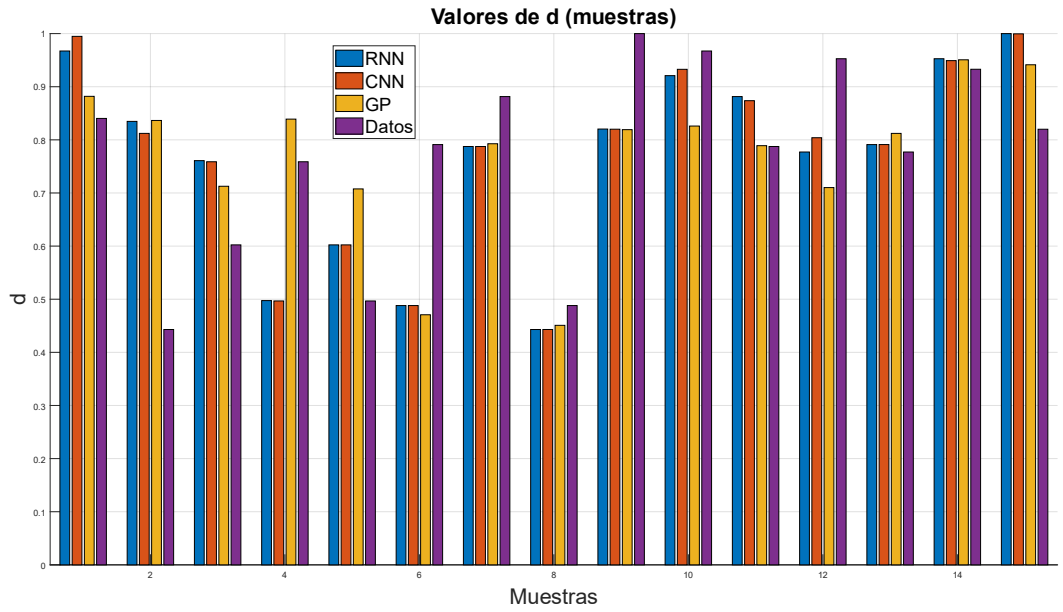


Figura 95: Valores obtenidos para cada modelo. Exponente de la función estrés efectivo del sólido.

A.3 Función de dosificación

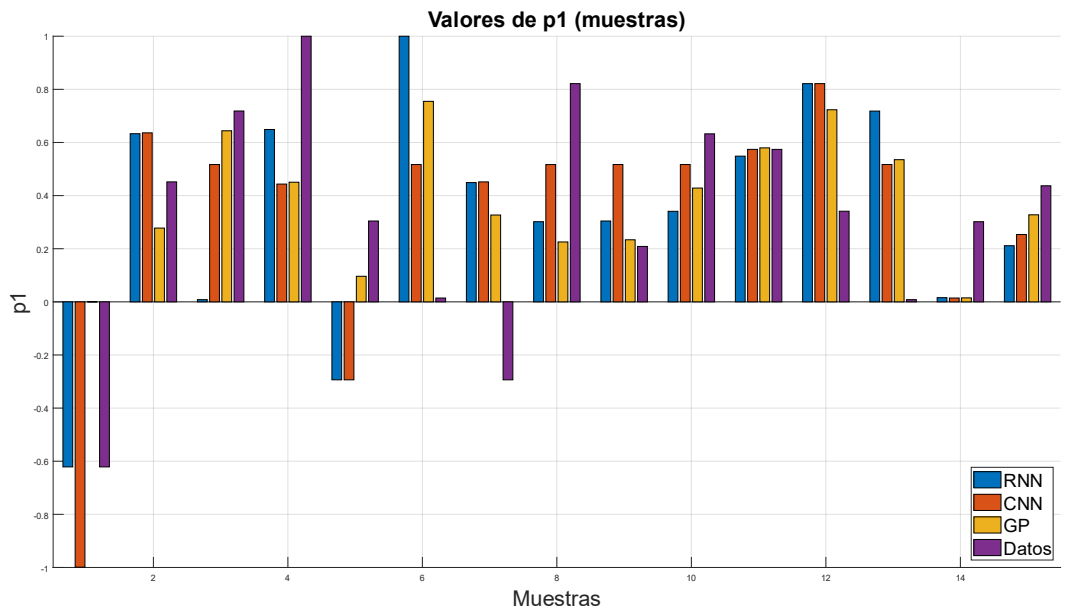


Figura 96: Valores obtenidos para cada modelo. Coeficiente cuadrático de la función de dosificación.

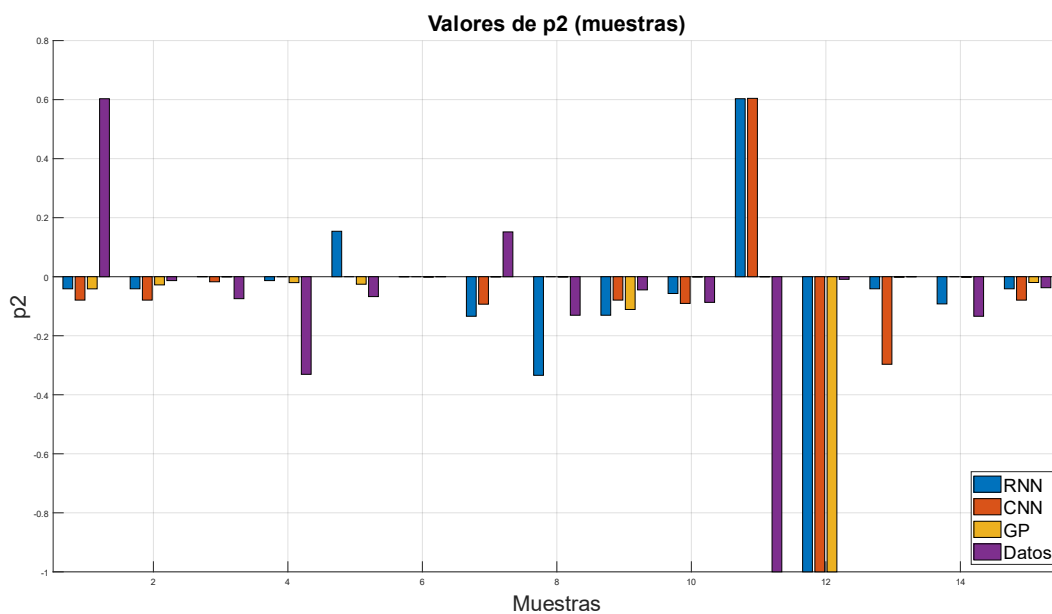


Figura 97: Valores obtenidos para cada modelo. Coeficiente lineal de la función de dosificación.

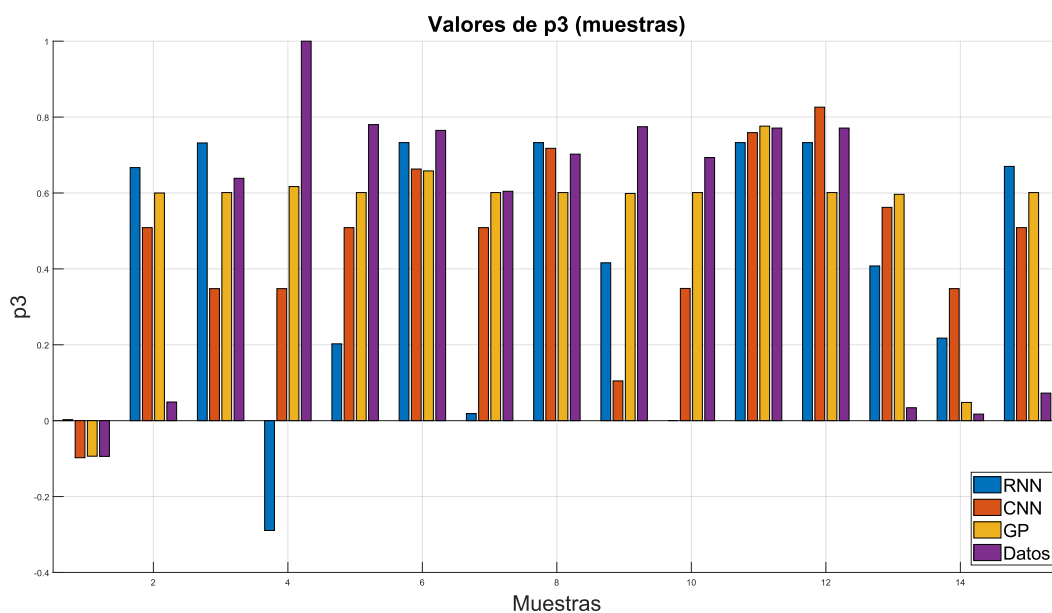


Figura 98: Valores obtenidos para cada modelo. Coeficiente libre de la función de dosificación.

A.4 Concentración crítica

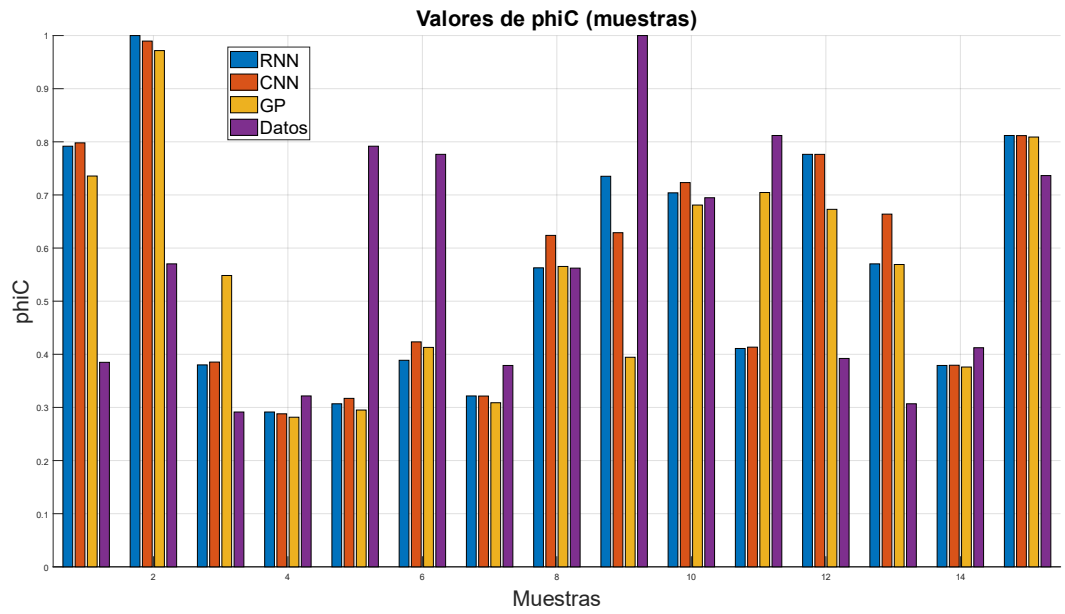


Figura 99: Valores obtenidos para cada modelo. Concentración crítica.

Anexo C: Funciones constitutivas para cada muestra de relave

B.1 Muestra R01

Función batch de Kynch

En la Figura 100 se observa la función batch de Kynch para la muestra R01, en función de la concentración. Para esta muestra, los modelos diseñados a partir de redes neuronales coinciden de mejor manera que el modelo de programación genética, donde se distancia principalmente en la zona de máxima convexidad de la función. Esto se debe a una subestimación tanto en el coeficiente A, donde se observa se observa un error porcentual del 8,08% para el algoritmo de programación genética (GP), 0,36% para la red neuronal recurrente (RNN) y 0,71% para la red neuronal convolucional (CNN), como para el coeficiente b, con un error porcentual para GP del 31,17%, para RNN un 3,03%, y 3,13% para CNN. Excluyendo el resultado obtenido mediante GP para el coeficiente b, todos los modelos se ajustan correctamente con la muestra, con rangos aceptables de error.

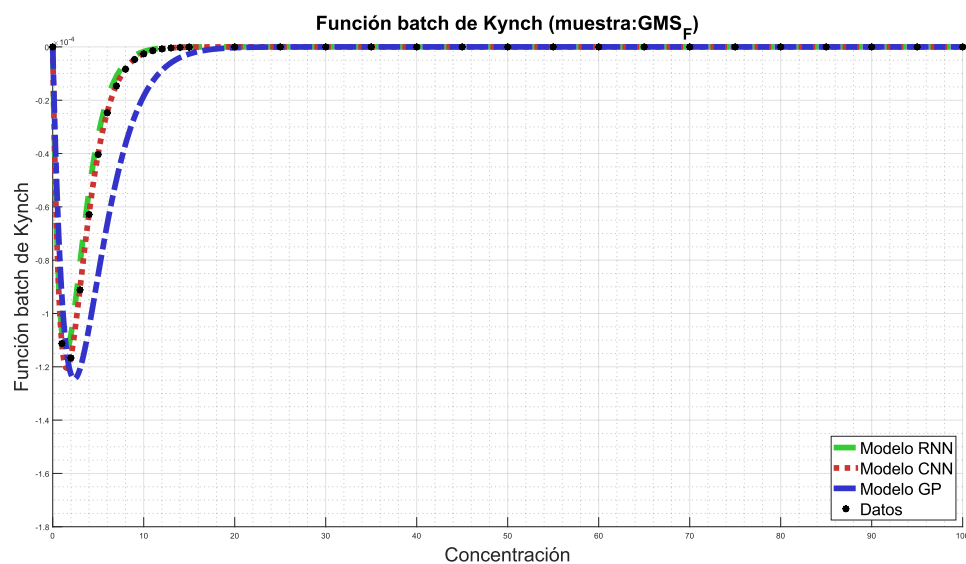


Figura 100: Simulación de la función batch de Kynch para la muestra R01.

Función de estrés efectivo del sólido

Para el caso de la muestra R01, en la Figura 101 se observa la curva descrita por la función de estrés efectivo del sólido. En la Figura se observa que para los 3 algoritmos se obtienen curvas que describen a los datos empíricos. Con respecto al coeficiente C, se obtuvo un error porcentual del 8,94% para GP, 1,27% para RNN y 1,11% para CNN. Con esto, se tiene un error aceptable para los 3 modelos estudiados. Por otro lado, el coeficiente d presenta un error porcentual del 7,49% para el modelo GP, 0,89% para el modelo RNN y 1,09% para el modelo CNN. Obteniéndose resultados similares para los 3 algoritmos, con respecto a lo observado en la práctica, como se observa en la Figura 101.

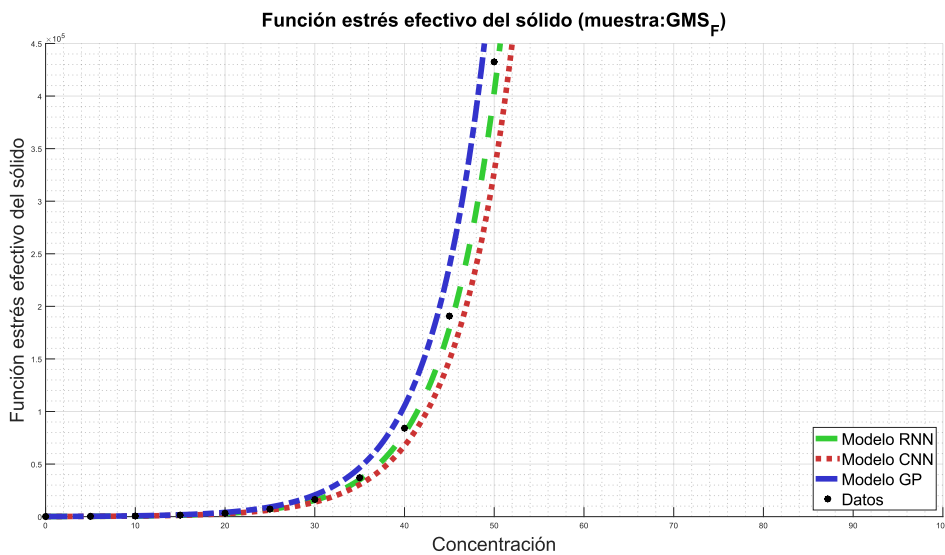


Figura 101: Simulación de la función estrés efectivo del sólido para la muestra R01.

Función de dosificación

En cuanto a la función de dosificación, en términos generales se observa en la Figura 102 que los modelos GP y CNN se aproximan bastante a la curva descrita por los datos, mientras que el modelo RNN se distancia, inclusive teniendo un signo contrario al descrito. Para el coeficiente cuadrático p_1 se tiene un error porcentual del 46,49% para el modelo GP, 9,75% para el

modelo RNN y 17,99% para el modelo CNN. Para el coeficiente lineal p_2 se observa un 41,23% de error para el modelo GP, 0,43% para el modelo RNN y 5,51% para el modelo CNN. Finalmente, para el coeficiente libre se tiene un error de 39,96% para el modelo GP, 199,79% para el modelo RNN, y 11,26% para el modelo CNN. Como análisis global de la función de dosificación se tiene que a medida que aumenta la concentración del relave, el modelo GP se va distanciando, mientras que el modelo RNN se mantiene a la misma distancia, al igual que el modelo CNN.

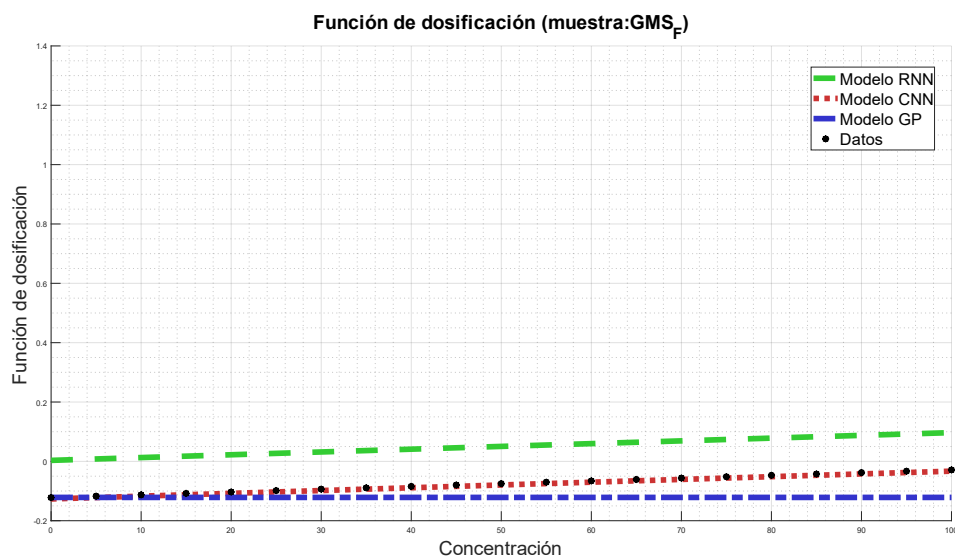


Figura 102: Simulación de la función de dosificación para la muestra R01.

Concentración crítica

En cuanto a la concentración crítica, los modelos que mejor reflejan el comportamiento de los datos son RN y CNN, mientras que el modelo GP se aleja de la predicción de éstos. El error porcentual del modelo GP es de 23,1%, mientras que para RNN y CNN es de 0,47% y 4,65%, respectivamente. En la Figura 103 se observa el valor de la concentración crítica para la muestra R01.



Figura 103: Simulación de la concentración crítica para la muestra R01.

B.2 Muestra R02

Función batch de Kynch

En la Figura 104 se observa la función batch de Kynch para la muestra R02, en función de la concentración. Para esta muestra, los modelos RNN y CNN vuelven a coincidir de mejor manera que el modelo GP, donde se distancia principalmente en la zona de máxima convexidad de la función. Para el coeficiente A se observa un error porcentual del 414,63% para el modelo GP, y 6,10% para los modelos RNN y CNN, mientras que para el coeficiente b, se obtuvo un error porcentual para el modelo GP del 14,58%, para el modelo RNN un 10,89%, y 3,50% para el modelo CNN. Excluyendo el resultado obtenido mediante GP para el coeficiente A, todos los modelos se ajustan correctamente con la muestra, con rangos aceptables de error.

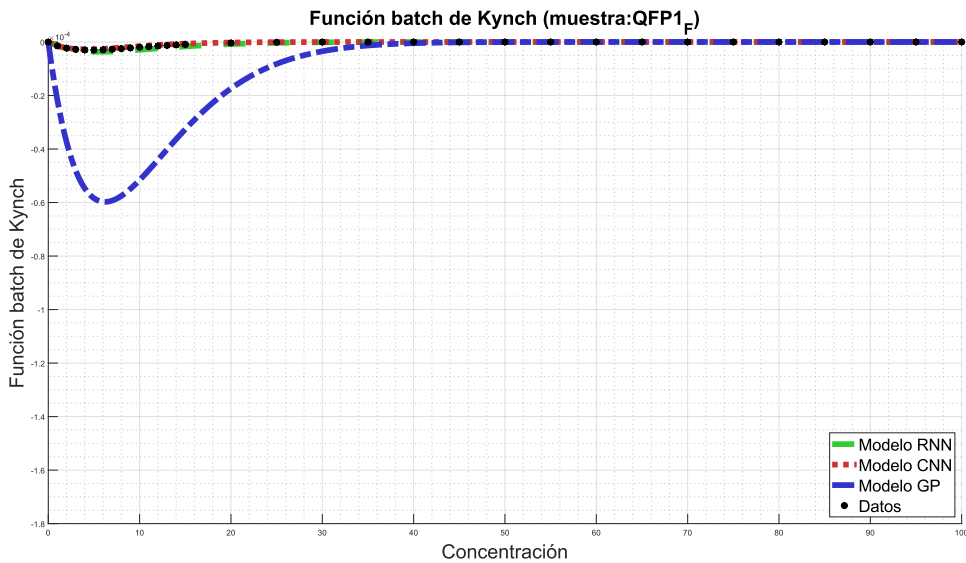


Figura 104: Simulación de la función batch de Kynch para la muestra R02.

Función de estrés efectivo del sólido

Para el caso de la muestra R01, en la Figura 105 se observa la curva descrita por la función de estrés efectivo del sólido. En la Figura se observa que para los 3 algoritmos se obtienen curvas que describen de buena manera a los datos empíricos. Con respecto al coeficiente C, se obtuvo un error porcentual del 0,52% para GP, 0,31% para el modelo RNN y 0% para el modelo CNN. Con esto, se tiene un error muy bajo para los 3 modelos estudiados. Por otro lado, el coeficiente d presenta un error porcentual del 3,23% para el modelo GP, 0,59% para el modelo RNN y 2,21% para el modelo CNN. Obteniéndose resultados similares para los 3 algoritmos, con respecto a lo observado en la práctica, como se observa en la Figura 105.

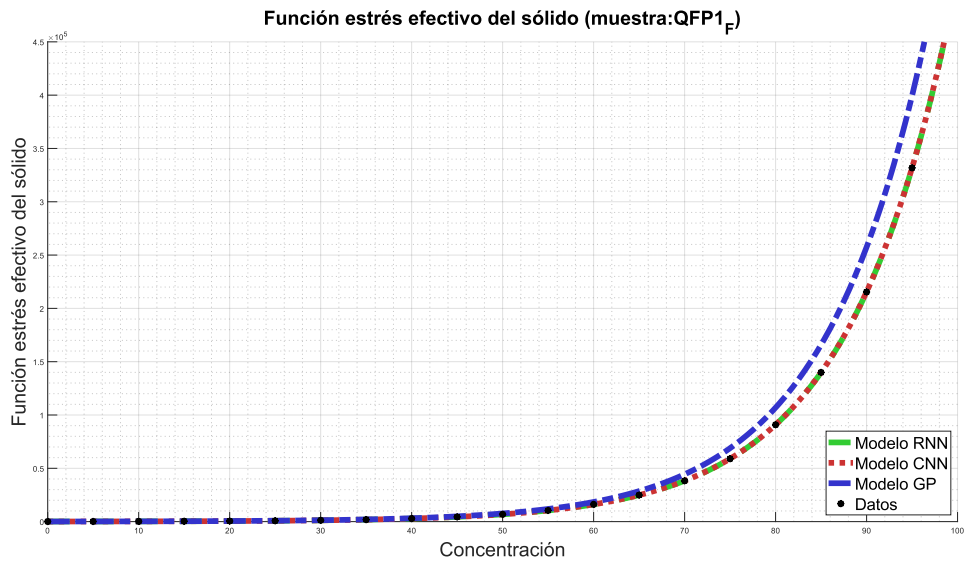


Figura 105: Simulación de la función estrés efectivo del sólido para la muestra R02.

Función de dosificación

En cuanto a la función de dosificación, en términos generales se observa en la Figura 106 que el modelo GP se ajusta a la curva descrita por los datos, mientras que los modelos RNN y CNN se distancian, manteniendo esa distancia a medida que aumenta la concentración. Para el coeficiente cuadrático p_1 se tiene un error porcentual del 6,62% para el modelo GP, 6,71% para el modelo RNN y 26,57% para el modelo CNN. Para el coeficiente lineal p_2 se observa un 228,03% de error para el modelo GP, 65,91% para el modelo RNN y 192,42% para el modelo CNN. Finalmente, para el coeficiente libre p_3 se tiene un error de 89,84% para el modelo GP, 71,75% para el modelo RNN, y 377,44% para el modelo CNN. Como análisis global de la función de dosificación para esta muestra, se observa un ajuste aceptable para el modelo GP, mientras que las redes neuronales se alejan de la predicción deseada.

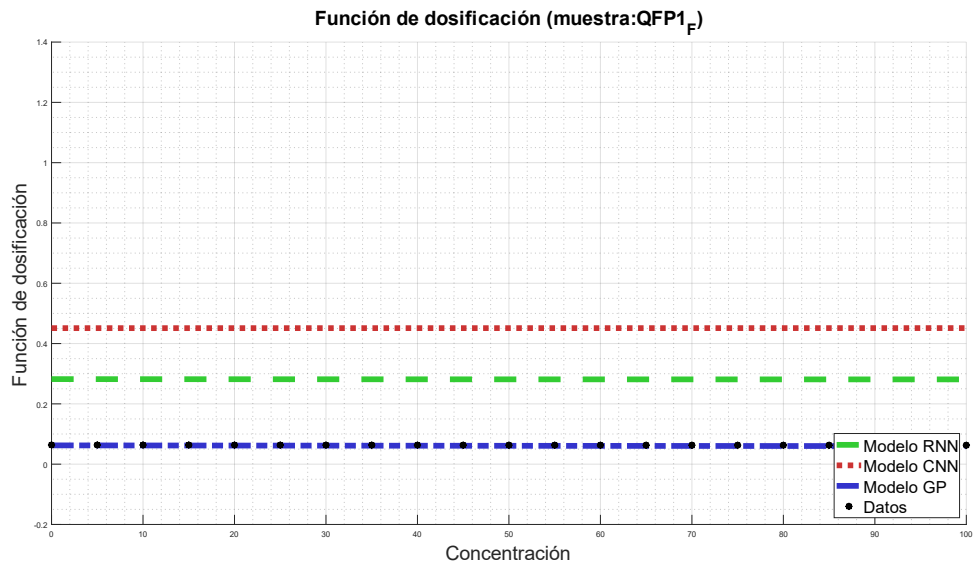


Figura 106: Simulación de la función de dosificación para la muestra R02.

Concentración crítica

En cuanto a la concentración crítica de esta muestra, los modelos que mejor reflejan el comportamiento de los datos son GP y RNN, mientras que el modelo CNN se aleja de la predicción de éstos. El error porcentual del modelo GP es de 1,19%, mientras que para RNN y CNN es de 2,39% y 2,21%, respectivamente. En la Figura 107 se observa el valor de la concentración crítica para la muestra R01.

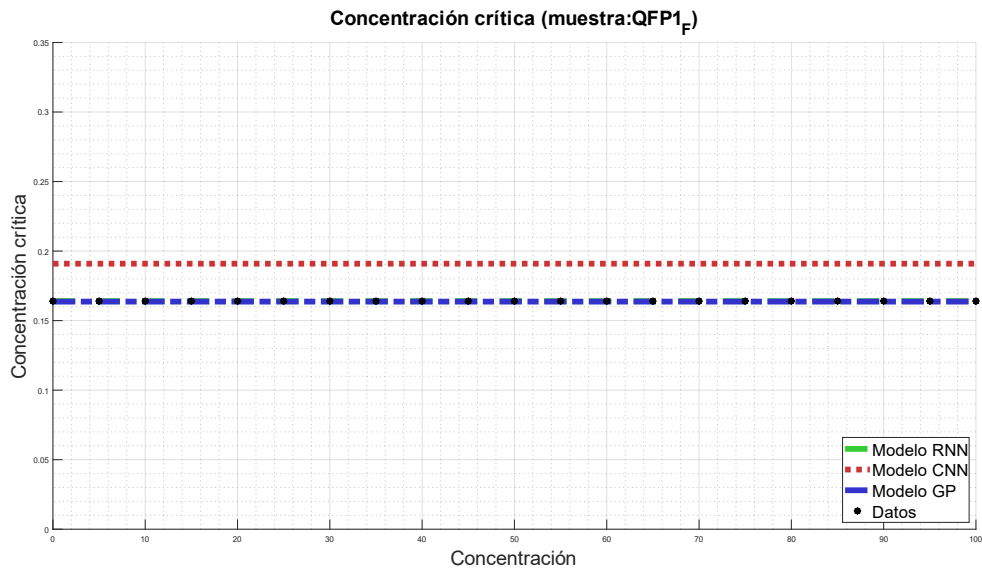


Figura 107: Simulación de la concentración crítica ϕ_c para la muestra R02.

B.3 Muestra R03

Función batch de Kynch

En la Figura 108 se observa la función batch de Kynch para la muestra R03, en función de la concentración. Para esta muestra, los modelos RNN y CNN vuelven a coincidir de mejor manera que el modelo GP, donde se distancia principalmente en la zona de máxima convexidad de la función. Para el coeficiente A se observa un error porcentual del 16,19% para el modelo GP, 0,19% para el modelo RNN y 0,14% para CNN, mientras que para el coeficiente b, se obtuvo un error porcentual para el modelo GP del 5,13%, para el modelo RNN un 1,73%, y 2,86% para el modelo CNN. Con esto, todos los modelos se ajustan correctamente con la muestra, con rangos aceptables de error.

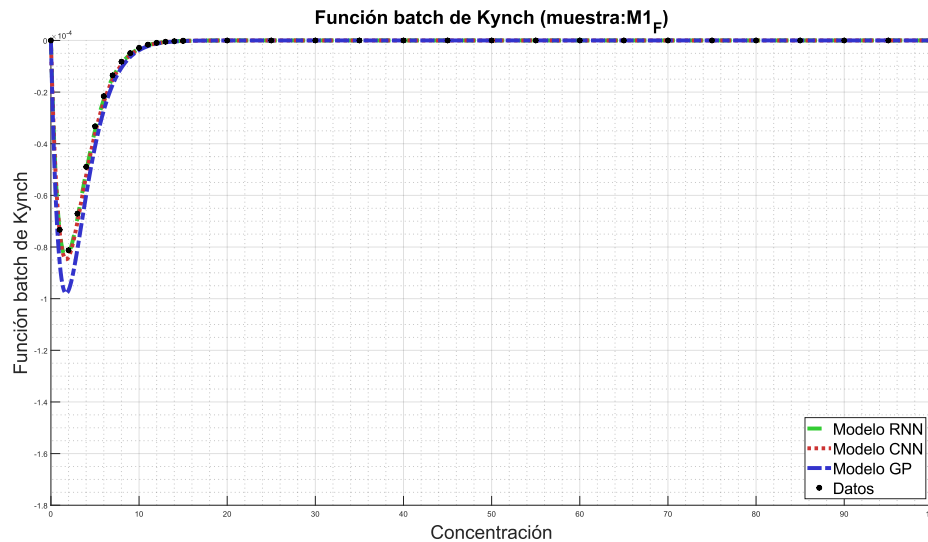


Figura 108: Simulación de la función batch de Kynch para la muestra R03.

Función de estrés efectivo del sólido

Para el caso de la muestra R03, en la Figura 109 se observa la curva descrita por la función de estrés efectivo del sólido. Se observa que para los modelos RNN y CNN se obtienen curvas que describen de buena manera a los datos empíricos, en especial este último. Con respecto al coeficiente C, se obtuvo un error porcentual del 1,16% para GP, 0,27% para el modelo RNN y 0,20% para el modelo CNN. Con esto, se tiene un error muy bajo para los 3 modelos estudiados. Por otro lado, el coeficiente d presenta un error porcentual del 8,68% para el modelo GP, 3,44% para el modelo RNN y 0,02% para el modelo CNN.

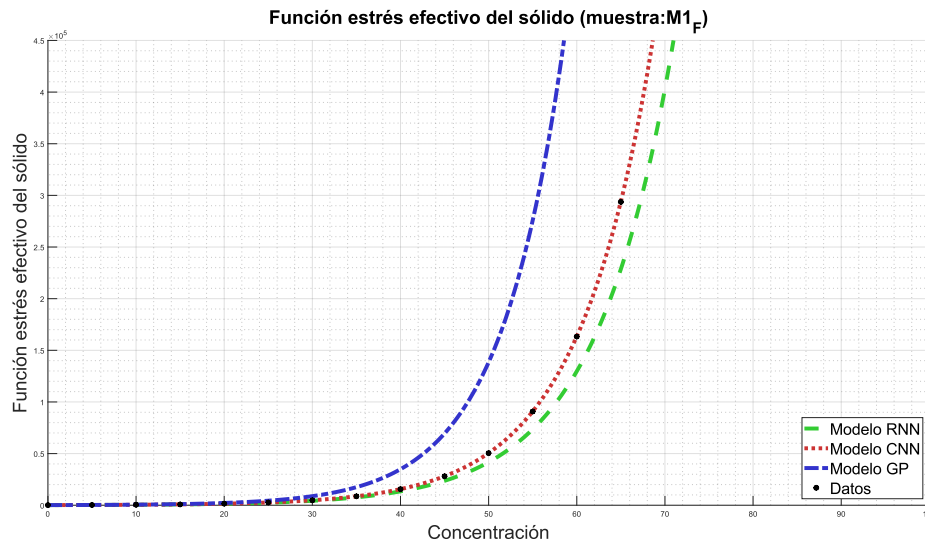


Figura 109: Simulación de la función estrés efectivo del sólido para la muestra R03.

Función de dosificación

En cuanto a la función de dosificación, en términos generales se observa en la Figura 110 que el modelo RNN se aproxima más a la curva descrita por los datos, mientras que los modelos GP y CNN se distancian, manteniendo esa distancia a medida que aumenta la concentración. Para el coeficiente cuadrático p_1 se tiene un error porcentual del 37,08% para el modelo GP, 10,69% para el modelo RNN y 9,13% para el modelo CNN. Para el coeficiente lineal p_2 se observa un 52,02% de error para el modelo GP, 24,19% para el modelo RNN y 63,31% para el modelo CNN. Finalmente, para el coeficiente libre p_3 se tiene un error de 24,86% para el modelo GP, 7,11% para el modelo RNN, y 25,10% para el modelo CNN. Como análisis global de la función de dosificación para esta muestra, se observa un ajuste aceptable para el modelo RNN, mientras que los demás se alejan de la predicción deseada.

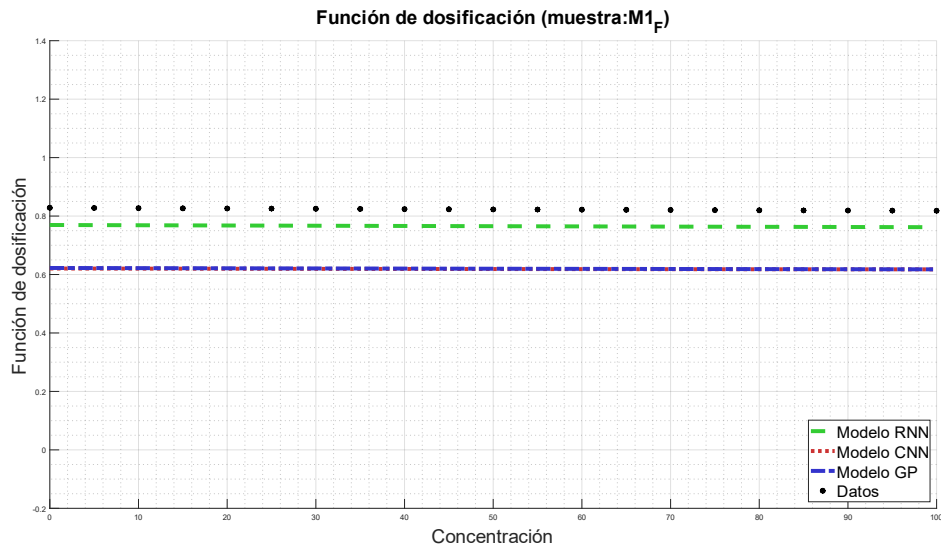


Figura 110: Simulación de la función de dosificación para la muestra R03.

Concentración crítica

En cuanto a la concentración crítica de esta muestra, todos los modelos representan el comportamiento de los datos. El error porcentual del modelo GP es de 4,05%, mientras que para RNN y CNN es de 8,00% y 11,32%, respectivamente. En la Figura 111 se observa el valor de la concentración crítica para la muestra R03.

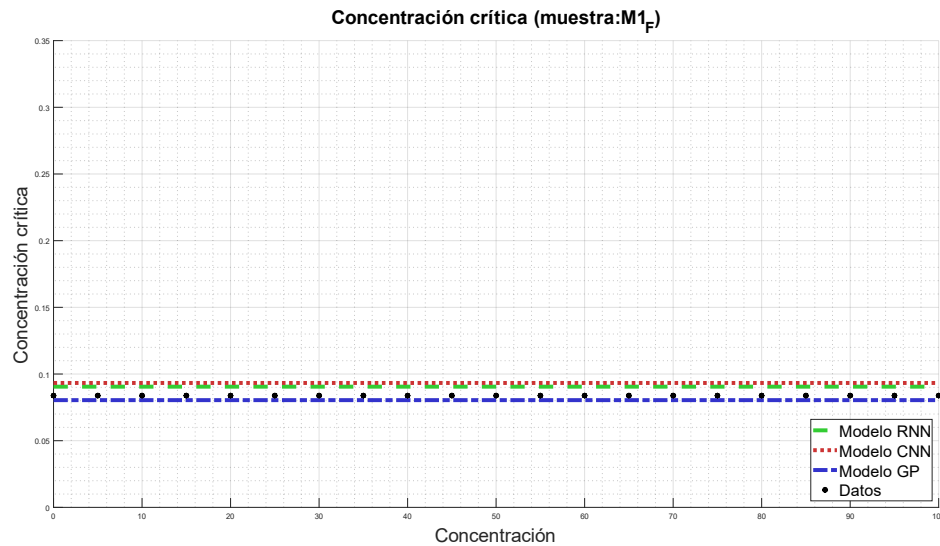


Figura 111: Simulación de la concentración crítica ϕ_c para la muestra R03.

B.4 Muestra R04

Función batch de Kynch

En la Figura 112 se observa la función batch de Kynch para la muestra R04, en función de la concentración. Para esta muestra, todos los modelos RNN y CNN coinciden de buena manera con los datos. Para el coeficiente A se observa un error porcentual del 10,59% para el modelo GP, 1,77% para el modelo RNN y 1,65% para CNN, mientras que para el coeficiente b, se obtuvo un error porcentual para el modelo GP del 8,22%, para el modelo RNN un 4,07%, y 4,33% para el modelo CNN. Para este caso, todos los modelos se ajustan correctamente con la muestra, con rangos aceptables de error.

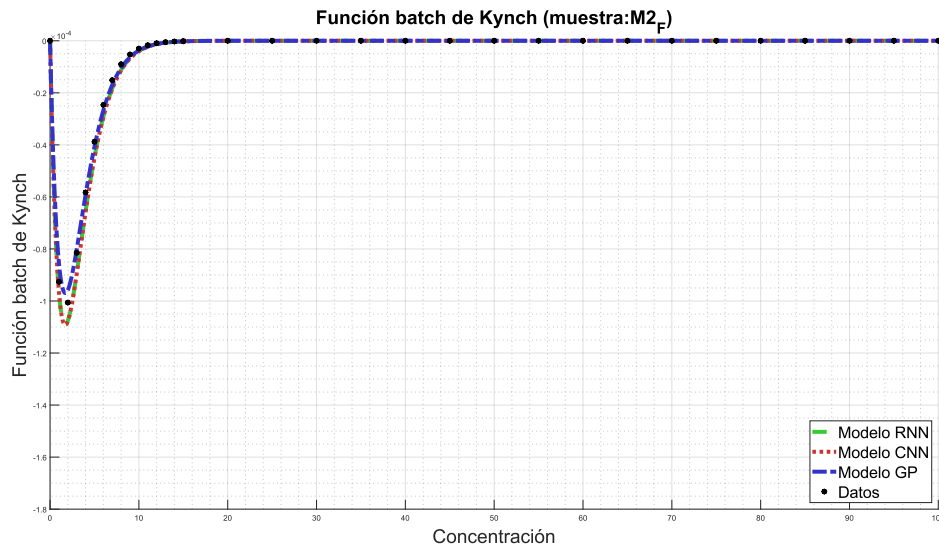


Figura 112: Simulación de la función batch de Kynch para la muestra R04.

Función de estrés efectivo del sólido

Para el caso de la muestra R04, en la Figura 113 se observa la curva descrita por la función de estrés efectivo del sólido. Se observa que para los modelos RNN y CNN se describen de buena manera a los datos empíricos, mientras que el modelo GP se aleja de éstos. Con respecto al coeficiente C, se obtuvo un error porcentual del 0,97% para GP, 0,04% para el modelo RNN y 0,08% para el modelo CNN. Por otro lado, el coeficiente d presenta un error porcentual del 2,98% para el modelo GP, 0,25% para el modelo RNN y 0,30% para el modelo CNN. Con esto, se tiene un error muy bajo para los 3 modelos estudiados.

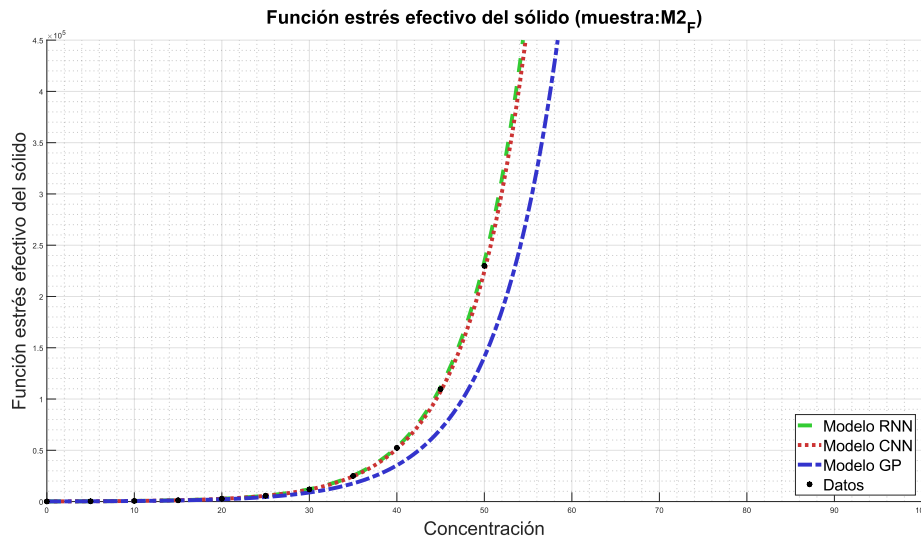


Figura 113: Simulación de la función estrés efectivo del sólido para la muestra R04.

Función de dosificación

En cuanto a la función de dosificación, en términos generales se observa en la Figura 114 que ningún modelo presenta un estimado aceptable para esta función, siendo el modelo RNN el más cercano con respecto a los datos. Para el coeficiente cuadrático p_1 se tiene un error porcentual del 47,36% para el modelo GP, 17,01% para el modelo RNN y 13,92% para el modelo CNN. Para el coeficiente lineal p_2 se observa un 89,30% de error para el modelo GP, 0,30% para el modelo RNN y 47,44% para el modelo CNN. Finalmente, para el coeficiente libre p_3 se tiene un error de 51,73% para el modelo GP, 8,55% para el modelo RNN, y 27,27% para el modelo CNN. Como análisis global de la función de dosificación para esta muestra, se observa un ajuste aceptable para el modelo RNN, mientras que los otros modelos se alejan notoriamente de la predicción deseada.

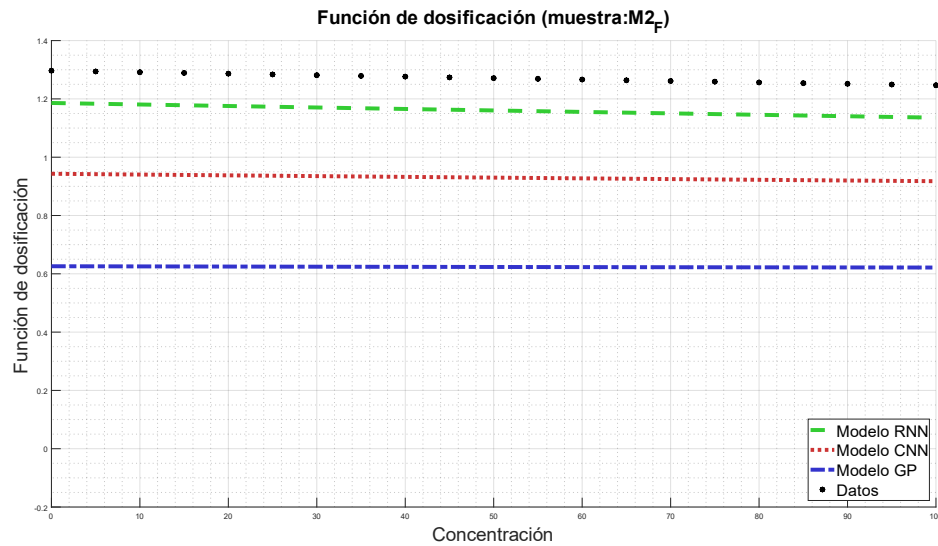


Figura 114: Simulación de la función de dosificación para la muestra R04.

Concentración crítica

En cuanto a la concentración crítica de esta muestra, los 3 modelos presentan un valor similar al dato real, siendo todos modelos aceptables para esta muestra. El error porcentual del modelo GP es de 5,75%, mientras que para RNN y CNN es de 0,25% y 0,44%, respectivamente. En la Figura 115 se observa el valor de la concentración crítica para la muestra R04.

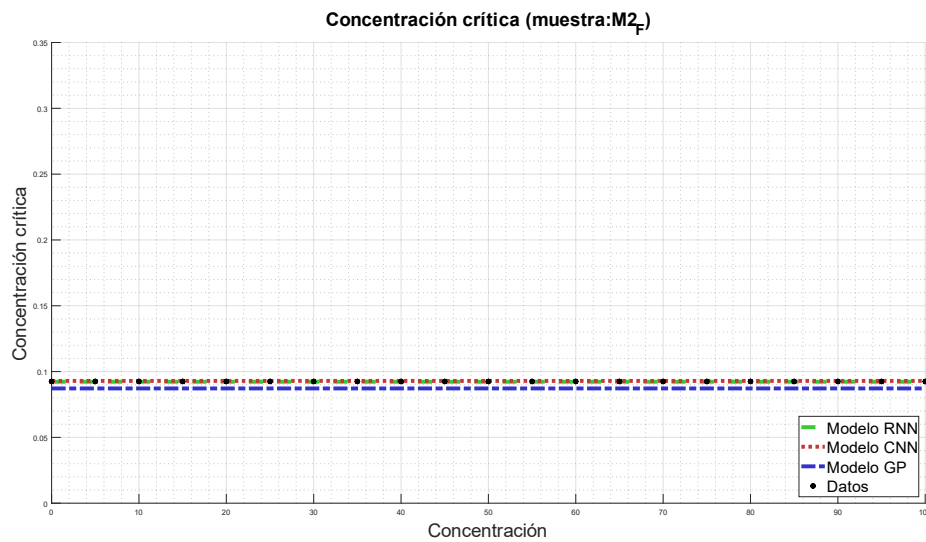


Figura 115: Simulación de la concentración crítica para la muestra R04.

B.5 Muestra R05

Función batch de Kynch

En la Figura 116 se observa la función batch de Kynch para la muestra R05, en función de la concentración. Para esta muestra, los modelos RNN y CNN vuelven a coincidir de mejor manera que el modelo GP, en especial RNN. Para el coeficiente A se observa un error porcentual del 281,50% para el modelo GP, y 0,25% para los modelos RNN y 19,5% para CNN, mientras que para el coeficiente b, se obtiene para el modelo GP un error del 20,51%, para el modelo RNN un 16,71%, y 10,72% para el modelo CNN. Excluyendo el resultado obtenido mediante GP para el coeficiente A, todos los modelos se ajustan correctamente con la muestra, con rangos aceptables de error.

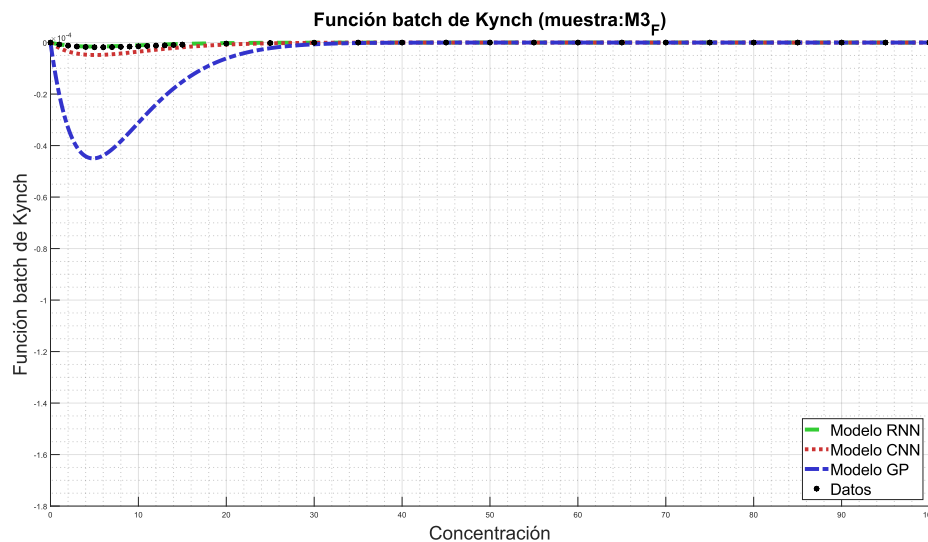


Figura 116: Simulación de la función batch de Kynch para la muestra R05.

Función de estrés efectivo del sólido

Para el caso de la muestra R05, en la Figura 117 se observa la curva descrita por la función de estrés efectivo del sólido. Se observa que para los modelos RNN y CNN coinciden correctamente con lo mostrado por el valor real, mientras que el modelo GP dista mucho de lo mostrado por los datos. Con respecto al coeficiente C, se obtuvo un error porcentual del 15,28% para GP, 0,36% para el modelo RNN y 0,29% para el modelo CNN. Con esto, se tiene un error muy bajo para los modelos RNN y CNN. Por otro lado, el coeficiente d presenta un error porcentual del 70,47% para el modelo GP, 0,30% para el modelo RNN y 0,041% para el modelo CNN. Obteniéndose resultados aceptables para los modelos de res neuronales, con respecto a lo presentado en la Figura 117.

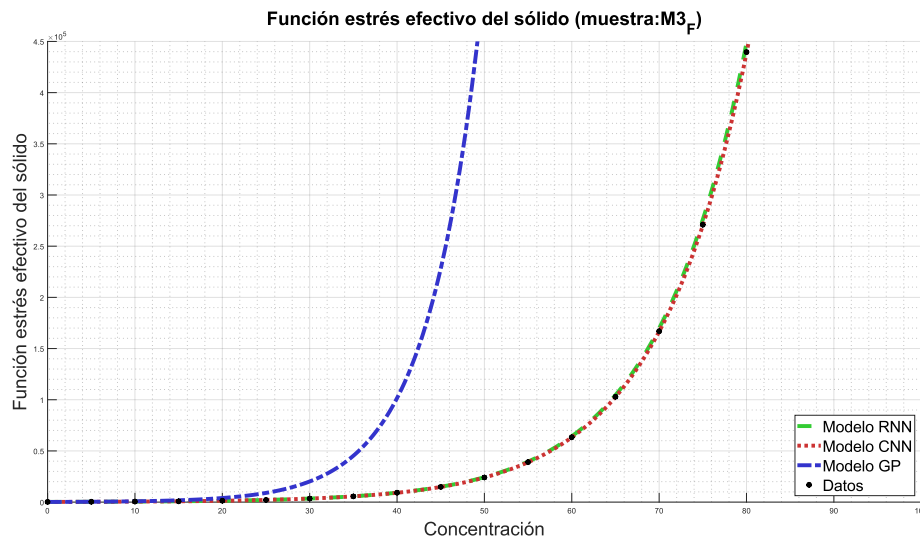


Figura 117: Simulación de la función estrés efectivo del sólido para la muestra R05.

Función de dosificación

En cuanto a la función de dosificación, en términos generales se observa en la Figura 118 que el modelo RNN se ajusta a la curva descrita por los datos, mientras que los modelos CNN y GP se distancian, manteniendo esa distancia a medida que aumenta la concentración. Para el coeficiente

cuadrático p_1 se tiene un error porcentual del 6,77% para el modelo GP, 24,80% para el modelo RNN y 21,25% para el modelo CNN. Para el coeficiente lineal p_2 se observa un 47,48% de error para el modelo GP, 0,89% para el modelo RNN y 26,56% para el modelo CNN. Finalmente, para el coeficiente libre p_3 se tiene un error de 28,25% para el modelo GP, 2,78% para el modelo RNN, y 18,66% para el modelo CNN. Como análisis global de la función de dosificación para esta muestra, se observa un ajuste aceptable para el modelo RNN, mientras que los demás modelos se alejan de la predicción deseada.

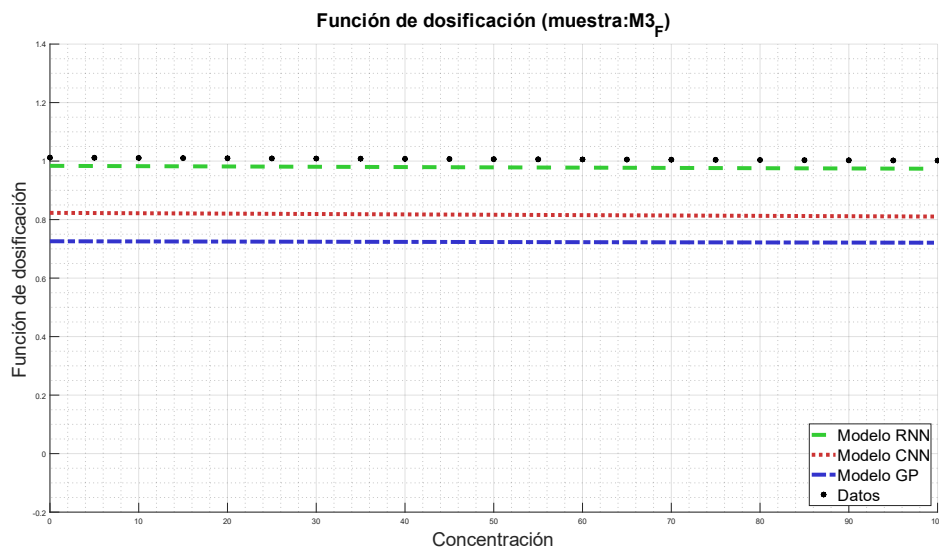


Figura 118: Simulación de la función de dosificación para la muestra R05.

Concentración crítica

En cuanto a la concentración crítica de esta muestra, el modelo que mejor refleja el comportamiento de los datos es RNN, mientras que el modelo GP y CNN están más distante de la predicción de éstos. El error porcentual del modelo GP es de 13,68%, mientras que para RNN y CNN es de 2,92% y 6,28%, respectivamente. En la Figura 119 se observa el valor de la concentración crítica para la muestra R05.

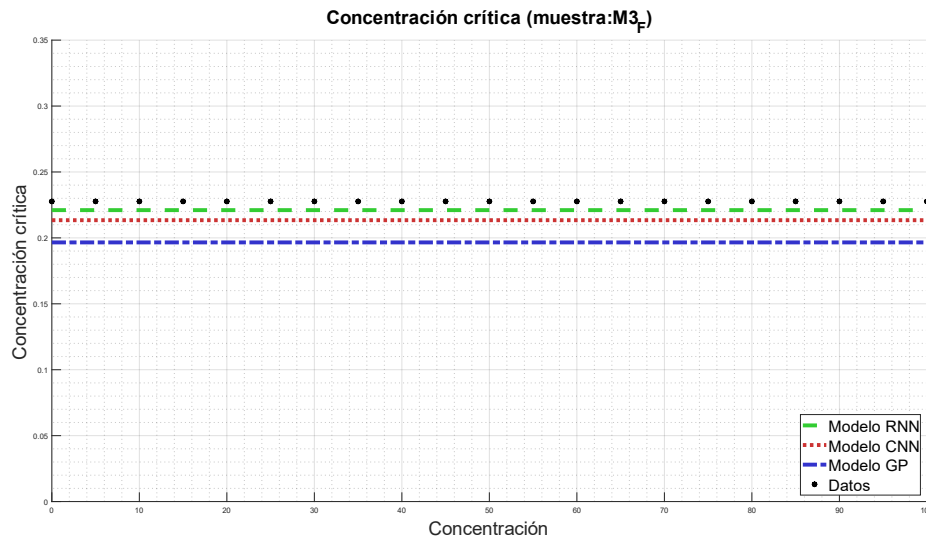


Figura 119: Simulación de la concentración crítica para la muestra R05.

B.6 Muestra R06

Función batch de Kynch

En la Figura 120 se observa la función batch de Kynch para la muestra R06, en función de la concentración. Para esta muestra, los modelos RNN y CNN (principalmente el primero) vuelven a coincidir de mejor manera que el modelo GP, donde se distancia principalmente en la zona de máxima convexidad de la función. Para el coeficiente A se observa un error porcentual del 356,94% para el modelo GP, y 1,74% para el modelo RNN y 42,71% para el modelo CNN, mientras que para el coeficiente b, se obtuvo un error porcentual para el modelo GP del 21,11%, para el modelo RNN un 2,01%, y 6,40% para el modelo CNN. Excluyendo el resultado obtenido mediante GP para el coeficiente A, todos los modelos se ajustan correctamente con la muestra, con rangos aceptables de error.

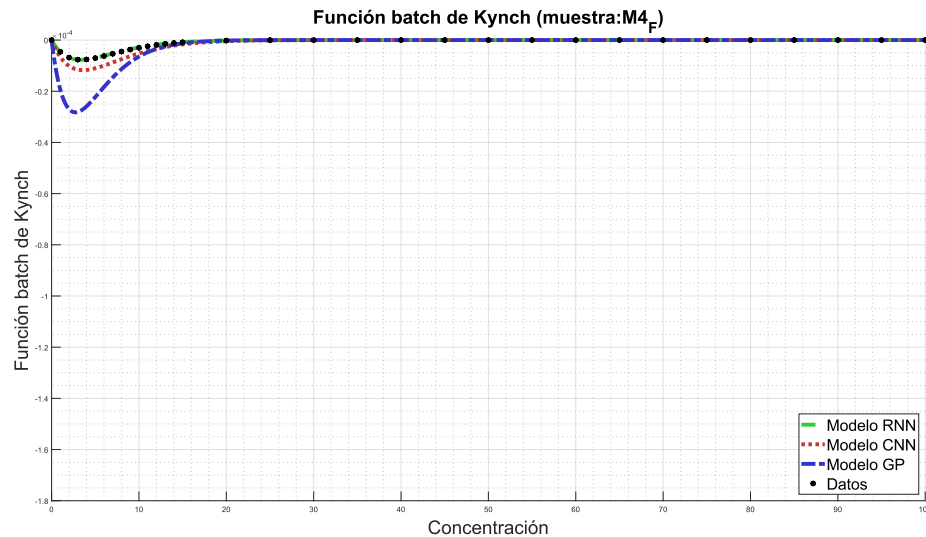


Figura 120: Simulación de la función batch de Kynch para la muestra R06.

Función de estrés efectivo del sólido

Para el caso de la muestra R01, en la Figura 121 se observa la curva descrita por la función de estrés efectivo del sólido. En la Figura se observa que para los 3 algoritmos se obtienen curvas que describen de buena manera a los datos empíricos, aunque la curva que mejor representa a los datos corresponde a la descrita por el modelo CNN, seguido por el modelo RNN y más lejano el modelo GP. Con respecto al coeficiente C , se obtuvo un error porcentual del 24,99% para GP, 0,59% para el modelo RNN y 0,06% para el modelo CNN. Con esto, se tiene un error muy bajo para los últimos dos modelos estudiados. Por otro lado, el coeficiente d presenta un error porcentual del 3,17% para el modelo GP, y 0,64% para los modelos RNN y CNN. Obteniéndose resultados similares para los 3 algoritmos, con respecto a lo observado en la práctica, como se observa en la Figura 121.

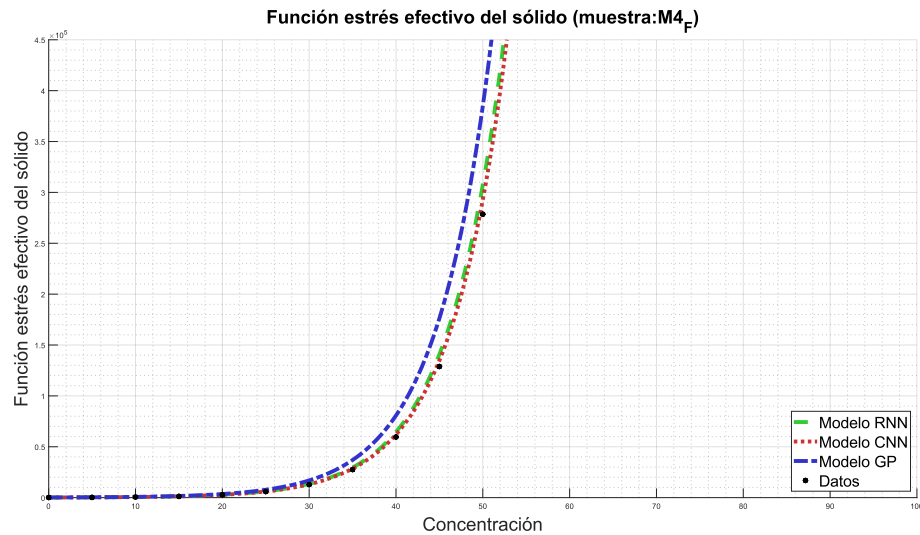


Figura 121: Simulación de la función estrés efectivo del sólido para la muestra R06.

Función de dosificación

En cuanto a la función de dosificación, en términos generales se observa en la Figura 122 que todos los modelos muestran valores distantes a la curva de datos experimentales. Para el coeficiente cuadrático p_1 se tiene un error porcentual del 728,47% para el modelo GP, 614,58% para el modelo RNN y 173,61% para el modelo CNN. Para el coeficiente lineal p_2 se observa un 890% de error para el modelo GP, 50% para el modelo RNN y 250% para el modelo CNN. Finalmente, para el coeficiente libre p_3 se tiene un error de 29,27% para el modelo GP, 36,85% para el modelo RNN, y 20,67% para el modelo CNN. Como análisis global de la función de dosificación para esta muestra, se observa un ajuste aceptable para el modelo GP, mientras que las redes neuronales se alejan de la predicción deseada.

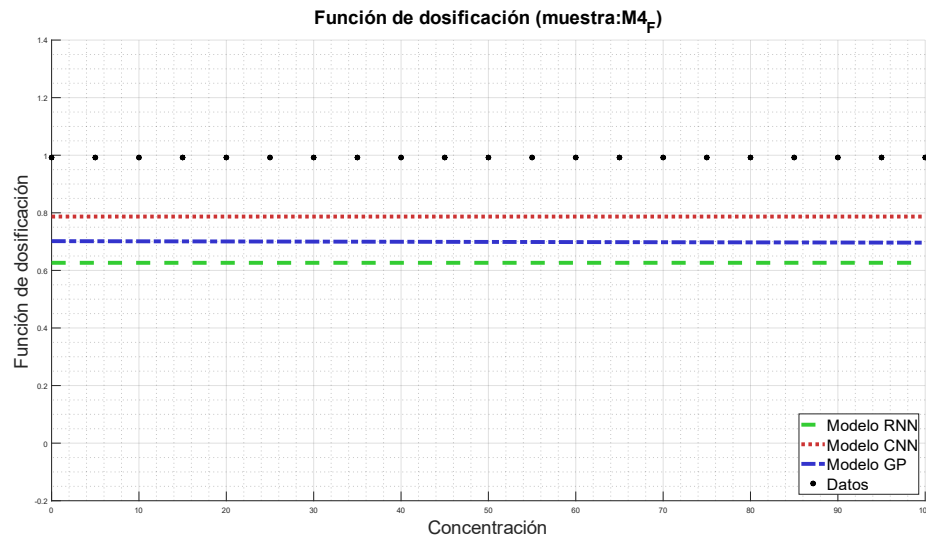


Figura 122: Simulación de la función de dosificación para la muestra R06.

Concentración crítica

En cuanto a la concentración crítica de esta muestra, los modelos que mejor reflejan el comportamiento de los datos son RNN, en menor medida el modelo CNN, y por último el modelo GP, quien se aleja de la predicción de éstos. El error porcentual del modelo GP es de 18,59%, mientras que para RNN y CNN es de 2,73% y 5,98%, respectivamente. En la Figura 123 se observa el valor de la concentración crítica para la muestra R06. En términos generales se tienen rangos aceptables de error, en especial para los modelos RNN y CNN.

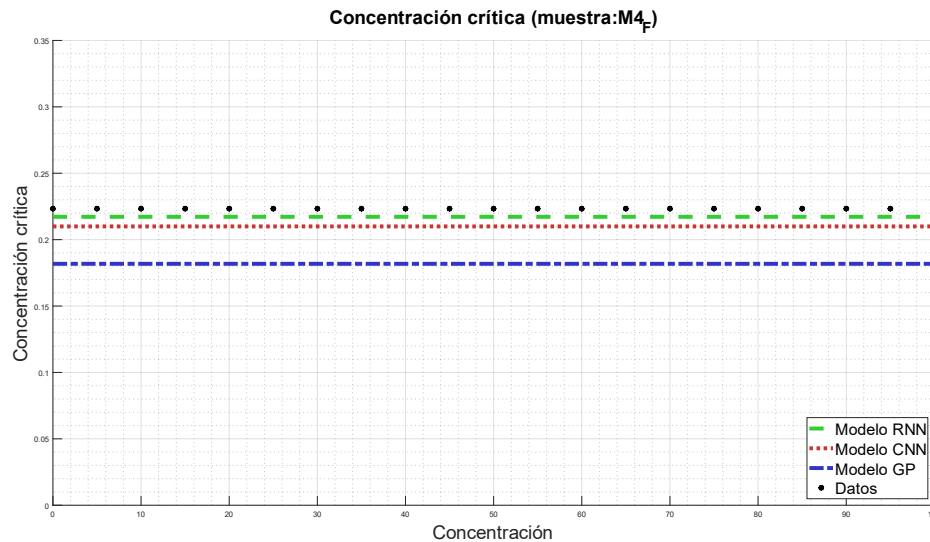


Figura 123: Simulación de la concentración crítica para la muestra R06.

B.7 Muestra R07

Función batch de Kynch

En la Figura 124 se observa la función batch de Kynch para la muestra R07, en función de la concentración. Para esta muestra, los modelos RNN y CNN coinciden de mejor manera que el modelo GP, donde se distancia principalmente en la zona de máxima convexidad de la función. Para el coeficiente A se observa un error porcentual del 1,04% para el modelo GP, y 0,20% para el modelo RNN y 0,48% para el modelo CNN, mientras que para el coeficiente b, se obtuvo un error porcentual para el modelo GP del 12,45%, para el modelo RNN un 0,47%, y 4,33% para el modelo CNN. Se puede notar para esta muestra que todos los modelos se ajustan correctamente, con rangos aceptables de error.

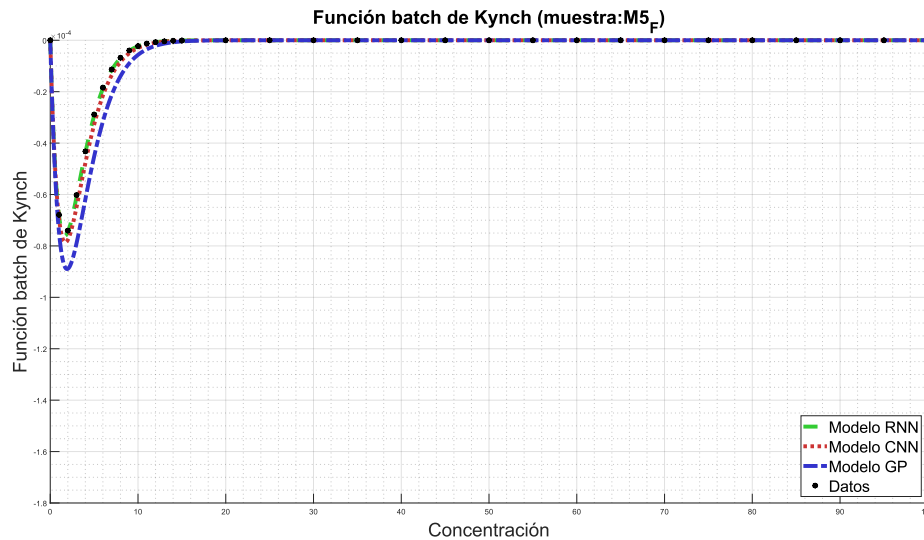


Figura 124: Simulación de la función batch de Kynch para la muestra R07.

Función de estrés efectivo del sólido

Para el caso de la muestra R01, en la Figura 125 se observa la curva descrita por la función de estrés efectivo del sólido. En la Figura se observa que para los modelos RNN y CNN se obtienen curvas que describen de buena manera a los datos empíricos, mientras que el modelo GP vuelve a alejarse a medida que aumenta la concentración. Con respecto al coeficiente C, se obtuvo un error porcentual del 1,03% para GP, 0,93% para el modelo RNN y 0,34% para el modelo CNN. Con esto, se tiene un error muy bajo para los 3 modelos estudiados. Por otro lado, el coeficiente d presenta un error porcentual del 4,42% para el modelo GP, 1,07% para el modelo RNN y 0,31% para el modelo CNN. Obteniéndose resultados similares para los 3 algoritmos, con respecto a lo observado en la práctica, como se observa en la Figura 125.

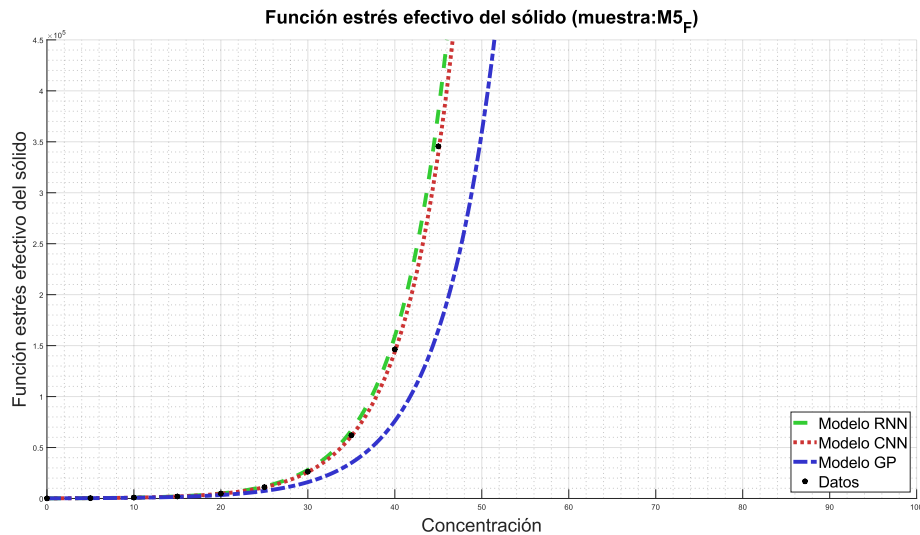


Figura 125: Simulación de la función estrés efectivo del sólido para la muestra R07.

Función de dosificación

En cuanto a la función de dosificación, en términos generales se observa en la Figura 126 que el modelo RNN se ajusta a la curva descrita por los datos, mientras que los modelos GP y CNN se distancian, manteniéndose alejados a medida que aumenta la concentración. Para el coeficiente cuadrático p_1 se tiene un error porcentual del 44,93% para el modelo GP, 20,06% para el modelo RNN y 40,87% para el modelo CNN. Para el coeficiente lineal p_2 se observa un 400,40% de error para el modelo GP, 245,19% para el modelo RNN y 5,80% para el modelo CNN. Finalmente, para el coeficiente libre p_3 se tiene un error de 13,00% para el modelo GP, 91,33% para el modelo RNN, y 10,87% para el modelo CNN. Como análisis global de la función de dosificación para esta muestra, se observa un ajuste aceptable para el modelo RNN, mientras que los demás modelos se alejan de la predicción deseada.

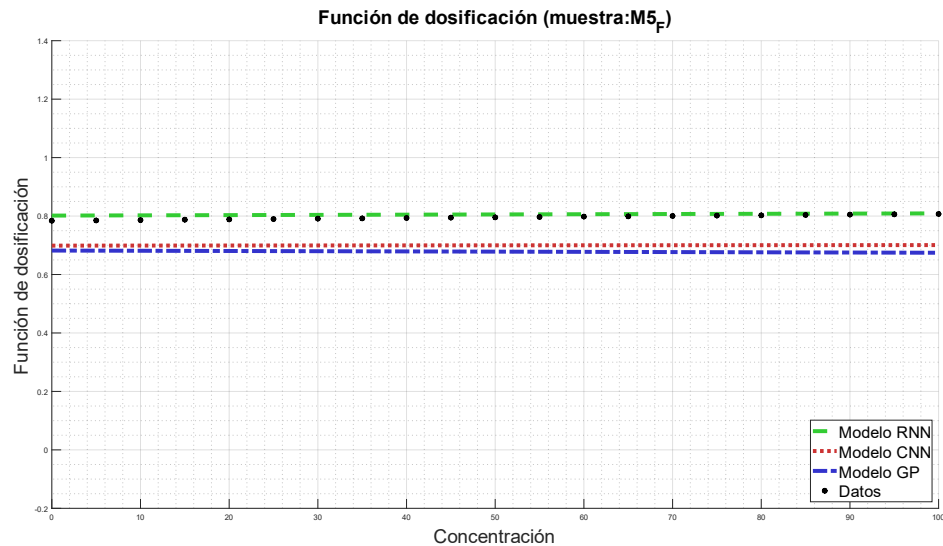


Figura 126: Simulación de la función de dosificación para la muestra R07.

Concentración crítica

En cuanto a la concentración crítica de esta muestra, los modelos que mejor reflejan el comportamiento de los datos son RNN y CNN, mientras que el modelo GP se aleja de la predicción de éstos. El error porcentual del modelo GP es de 18,98%, mientras que para RNN y CNN es de 3,96% y 7,05%, respectivamente. En la Figura 127 se observa el valor de la concentración crítica para la muestra R07.

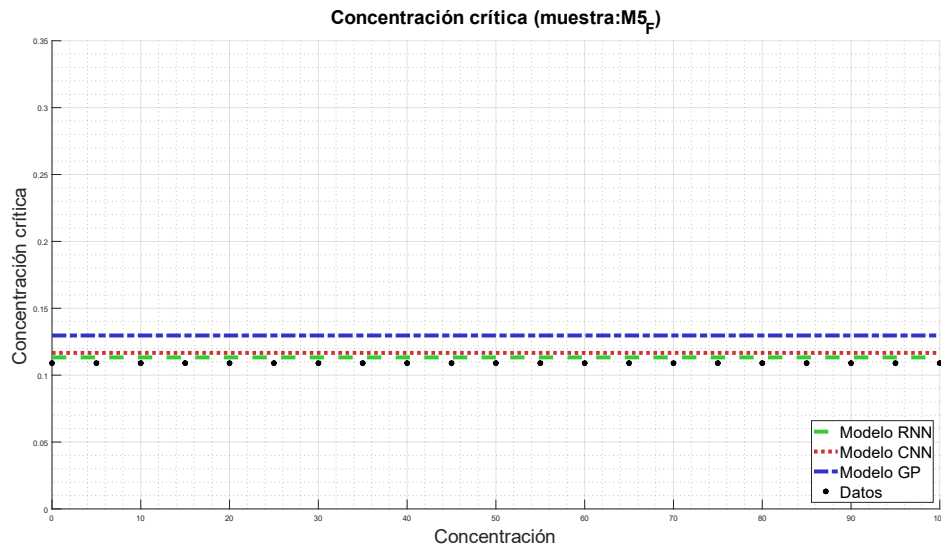


Figura 127: Simulación de la concentración crítica para la muestra R07.

B.8 Muestra R08

Función batch de Kynch

En la Figura 128 se observa la función batch de Kynch para la muestra R08, en función de la concentración. Para esta muestra, los modelos RNN y CNN coinciden de mejor manera que el modelo GP, donde se distancia principalmente en la zona de máxima convexidad de la función. Para el coeficiente A se observa un error porcentual del 88,73% para el modelo GP, y 0,81% para los modelos RNN y CNN, mientras que para el coeficiente b, se obtuvo un error porcentual para el modelo GP del 4,09%, para el modelo RNN un 0,16%, y 0,79% para el modelo CNN. Excluyendo el resultado obtenido mediante GP para el coeficiente A, todos los modelos se ajustan correctamente con la muestra, con rangos aceptables de error.

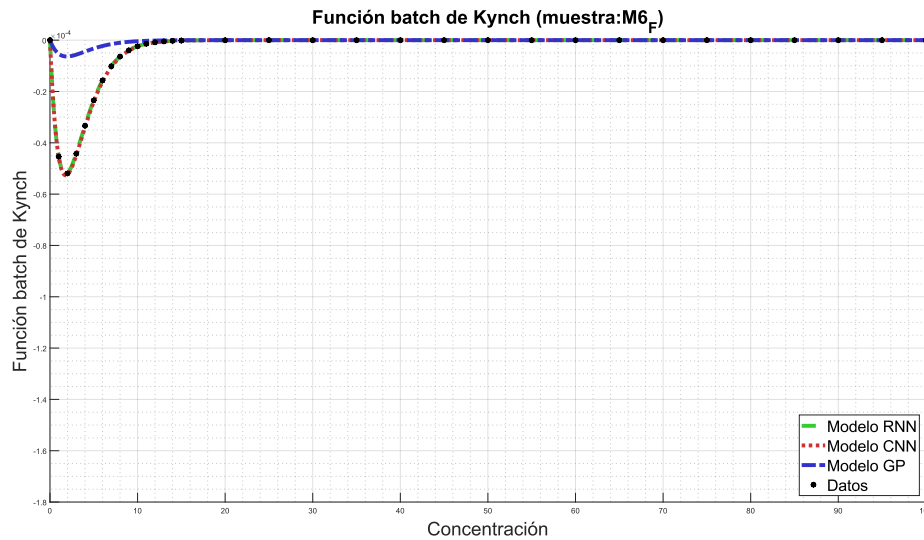


Figura 128: Simulación de la función batch de Kynch para la muestra R08.

Función de estrés efectivo del sólido

Para el caso de la muestra R01, en la Figura 129 se observa la curva descrita por la función de estrés efectivo del sólido. En la Figura se observa que para los 3 algoritmos se obtienen curvas que describen de buena manera a los datos empíricos. Con respecto al coeficiente C, se obtuvo un error porcentual del 1,81% para GP, 0,42% para el modelo RNN y 0,019% para el modelo CNN. Con esto, se tiene un error muy bajo para los 3 modelos estudiados. Por otro lado, el coeficiente d presenta un error porcentual del 0,59% para el modelo GP, 0,82% para el modelo RNN y 1,56% para el modelo CNN. Obteniéndose resultados similares para los 3 algoritmos, con respecto a lo observado en la práctica, como se observa en la Figura 129.

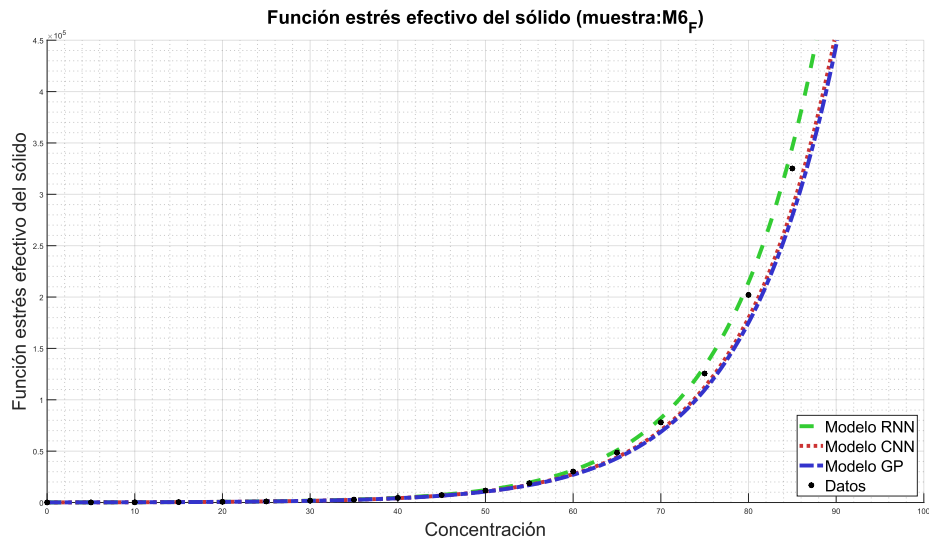


Figura 129: Simulación de la función estrés efectivo del sólido para la muestra R08.

Función de dosificación

En cuanto a la función de dosificación, en términos generales se observa en la Figura 130 que el modelo GP se ajusta a la curva descrita por los datos, mientras que los modelos RNN y CNN se distancian, manteniendo esa distancia a medida que aumenta la concentración. Para el coeficiente cuadrático p_1 se tiene un error porcentual del 9,30% para el modelo GP, 9,20% para el modelo RNN y 2,11% para el modelo CNN. Para el coeficiente lineal p_2 se observa un 38,06% de error para el modelo GP, 2,91% para el modelo RNN y 35,76% para el modelo CNN. Finalmente, para el coeficiente libre p_3 se tiene un error de 38,85% para el modelo GP, 21,01% para el modelo RNN, y 14,68% para el modelo CNN. Como análisis global de la función de dosificación para esta muestra, se observa que ninguno modelo representa los valores presentados por los datos experimentales, alejándose de su predicción.

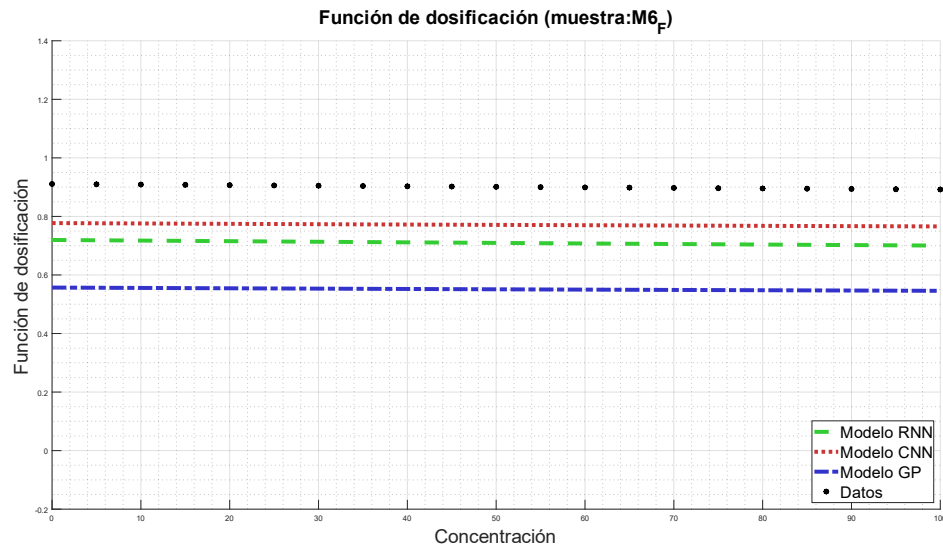


Figura 130: Simulación de la función de dosificación para la muestra R08.

Concentración crítica

En cuanto a la concentración crítica de esta muestra, todos los modelos representan de buena manera a la curva de datos, presentando valores en torno al real. El error porcentual del modelo GP es de 3,04%, mientras que para RNN y CNN es de 1,41% y 4,38%, respectivamente. En la Figura 131 se observa el valor de la concentración crítica para la muestra R08.

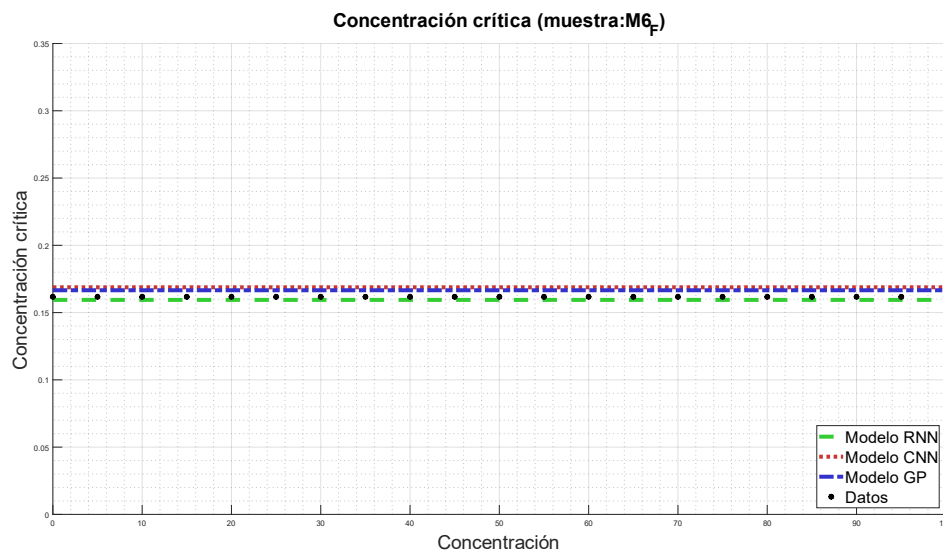


Figura 131: Simulación de la concentración crítica para la muestra R08.

B.9 Muestra R09

Función batch de Kynch

En la Figura 132 se observa la función batch de Kynch para la muestra R09, en función de la concentración. Para esta muestra, todos los modelos presentan de buena forma a la curva de los datos, siendo el modelo RNN el que mejor los representa. Para el coeficiente A se observa un error porcentual del 20,71% para el modelo GP, 0,06% para los modelos RNN, y 0,2% para el modelo CNN, mientras que para el coeficiente b, se obtuvo un error porcentual para el modelo GP del 12,44%, para el modelo RNN un 6,14%, y 6,67% para el modelo CNN. Excluyendo el resultado obtenido mediante GP para el coeficiente A, todos los modelos se ajustan correctamente con la muestra, con rangos aceptables de error.

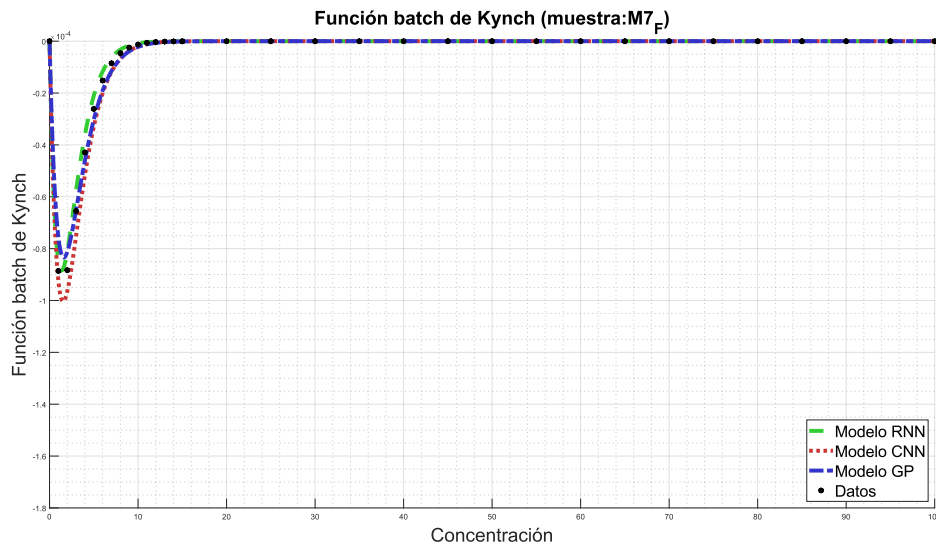


Figura 132: Simulación de la función batch de Kynch para la muestra R09.

Función de estrés efectivo del sólido

Para el caso de la muestra R09, en la Figura 133 se observa la curva descrita por la función de estrés efectivo del sólido. En la Figura se observa que para los 3 algoritmos se obtienen curvas que describen de buena manera a los

datos empíricos. Con respecto al coeficiente C , se obtuvo un error porcentual del 3,73% para GP, 0,51% para el modelo RNN y 0,46% para el modelo CNN. Con esto, se tiene un error muy bajo para los 3 modelos estudiados. Por otro lado, el coeficiente d presenta un error porcentual del 0,73% para el modelo GP, 0,01% para el modelo RNN y 0,03% para el modelo CNN. Obteniéndose resultados similares para los 3 algoritmos, con respecto a lo observado en la práctica, como se observa en la Figura 133.

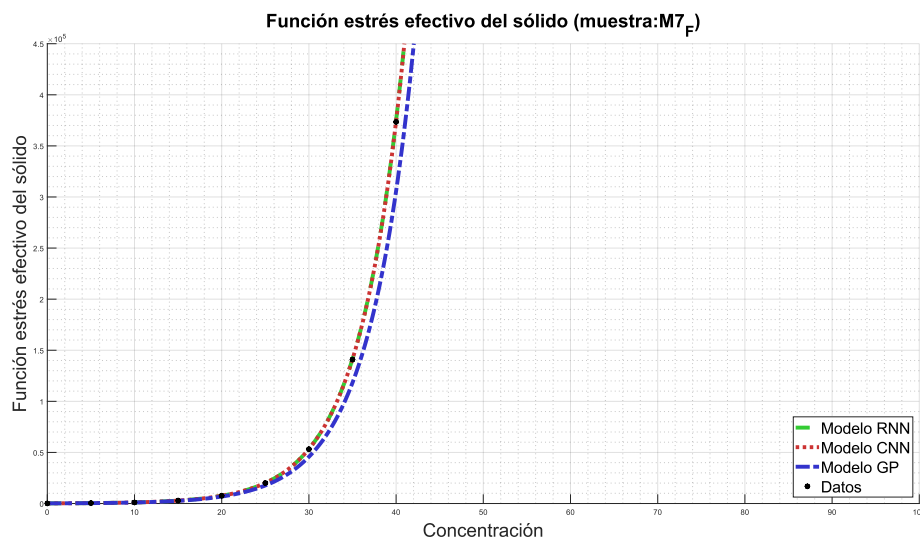


Figura 133: Simulación de la función estrés efectivo del sólido para la muestra R09.

Función de dosificación

En cuanto a la función de dosificación, en términos generales se observa en la Figura 134 que el modelo RNN se ajusta a la curva descrita por los datos, mientras que los modelos RNN y CNN se distancian, en especial este último. Para el coeficiente cuadrático p_1 se tiene un error porcentual del 101,39% para el modelo GP, 1,68% para el modelo RNN y 60,31% para el modelo CNN. Para el coeficiente lineal p_2 se observa un 3,14% de error para el modelo GP, 2,24% para el modelo RNN y 3,59% para el modelo CNN. Finalmente, para el coeficiente libre p_3 se tiene un error de 16,90% para el modelo GP, 1,25% para el modelo RNN, y 4,93% para el modelo CNN. Como análisis global de la función de dosificación para esta muestra,

se observa un ajuste aceptable para el modelo RNN, mientras que los demás modelos se alejan de la predicción deseada.

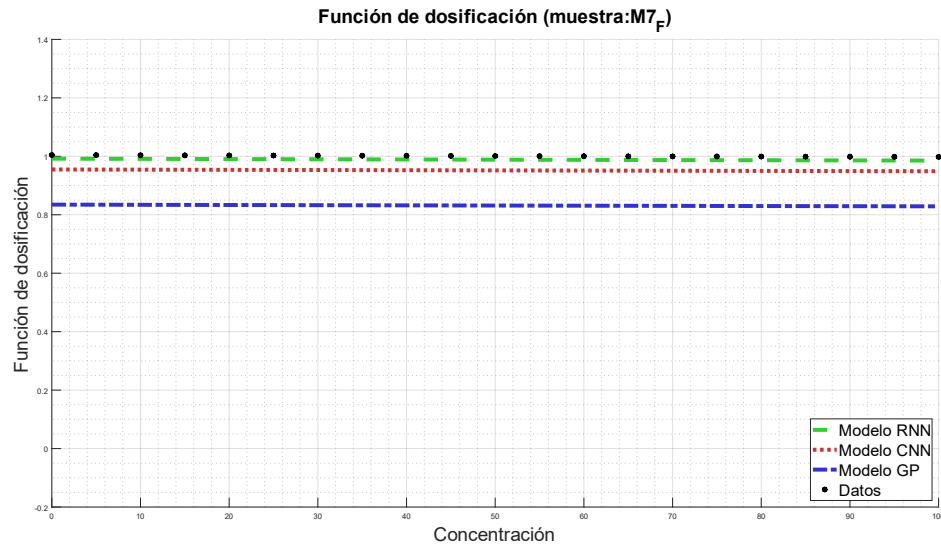


Figura 134: Simulación de la función de dosificación para la muestra R09.

Concentración crítica

En cuanto a la concentración crítica de esta muestra, los modelos que mejor reflejan el comportamiento de los datos son RNN y CNN, mientras que el modelo GP se aleja de la predicción de éstos. El error porcentual del modelo GP es de 13,73%, mientras que para RNN y CNN es de 0,30% y 1,24%, respectivamente. En la Figura 135 se observa el valor de la concentración crítica para la muestra R09.

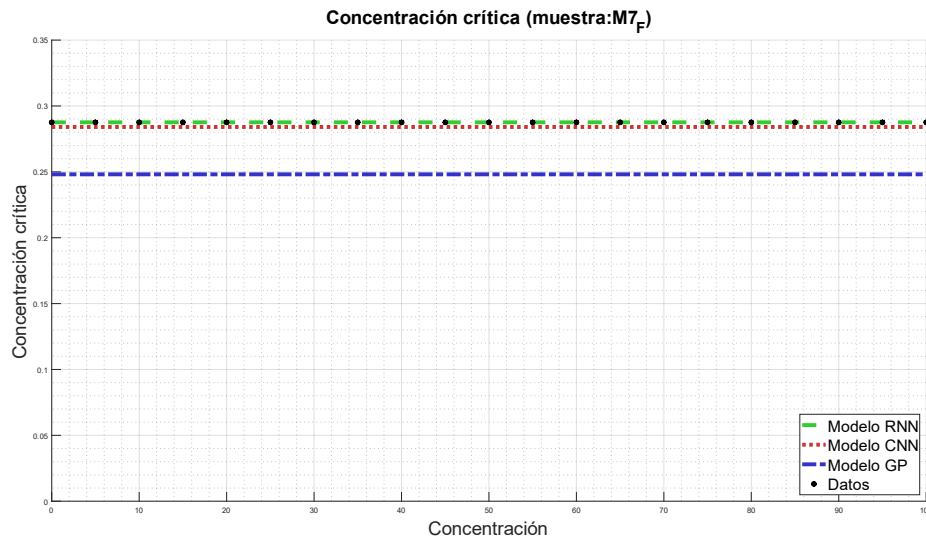


Figura 135: Simulación de la concentración crítica para la muestra R09.

B.10 Muestra R10

Función batch de Kynch

En la Figura 136 se observa la función batch de Kynch para la muestra R10, en función de la concentración. Para esta muestra, todos los modelos representan de buena manera a los datos experimentales, presentando bajos valores de error. Para el coeficiente A se observa un error porcentual del 4,20% para el modelo GP, 0,14% para los modelos RNN y 8,013% para el modelo CNN, mientras que para el coeficiente b, se obtuvo un error porcentual para el modelo GP del 0,68%, para el modelo RNN un 4,12%, y 5,65% para el modelo CNN. Se observa en la figura que las curvas coinciden en gran medida con la curva de los datos.

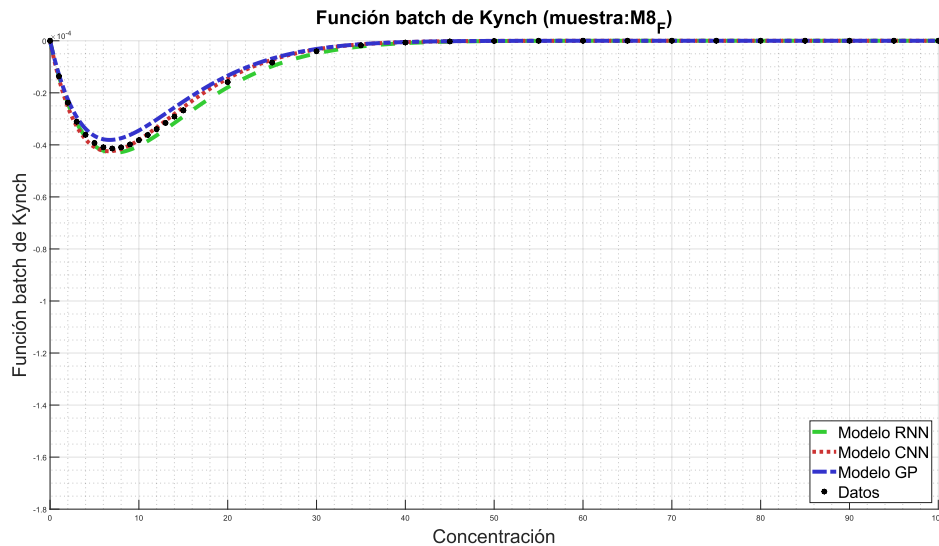


Figura 136: Simulación de la función batch de Kynch para la muestra R10.

Función de estrés efectivo del sólido

Para el caso de la muestra R10, en la Figura 137 se observa la curva descrita por la función de estrés efectivo del sólido. Se observa que para los 3 algoritmos se obtienen curvas que describen de buena manera a los datos empíricos, especialmente RNN y CNN. Con respecto al coeficiente C, se obtuvo un error porcentual del 1,02% para GP, 1,45% para el modelo RNN y 0,75% para el modelo CNN. Con esto, se tiene un error muy bajo para los 3 modelos estudiados. Por otro lado, el coeficiente d presenta un error porcentual del 2,16% para el modelo GP, 0,19% para el modelo RNN y 0,86% para el modelo CNN, obteniéndose resultados similares para los 3 algoritmos, con respecto a lo observado en la práctica, como se observa en la Figura 137.

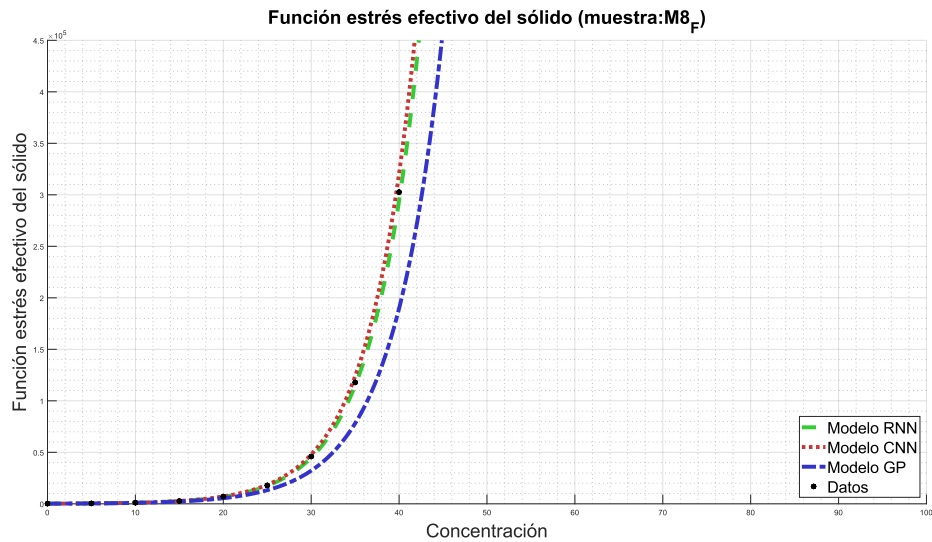


Figura 137: Simulación de la función estrés efectivo del sólido para la muestra R10.

Función de dosificación

En cuanto a la función de dosificación, en términos generales se observa en la Figura 138 que el modelo RNN se ajusta a la curva descrita por los datos, seguida por el modelo CNN, y la más alejada corresponde a la curva del modelo GP. Para el coeficiente cuadrático p_1 se tiene un error porcentual del 34,75% para el modelo GP, 4,24% para el modelo RNN y 3,27% para el modelo CNN. Para el coeficiente lineal p_2 se observa un 46,49% de error para el modelo GP, 20,25% para el modelo RNN y 50,75% para el modelo CNN. Finalmente, para el coeficiente libre p_3 se tiene un error de 9,23% para el modelo GP, 1,70% para el modelo RNN, y 3,66% para el modelo CNN. Como análisis global de la función de dosificación para esta muestra, se observa un ajuste aceptable para los modelos RNN y CNN, mientras que el algoritmo de programación genética se aleja de la predicción deseada.

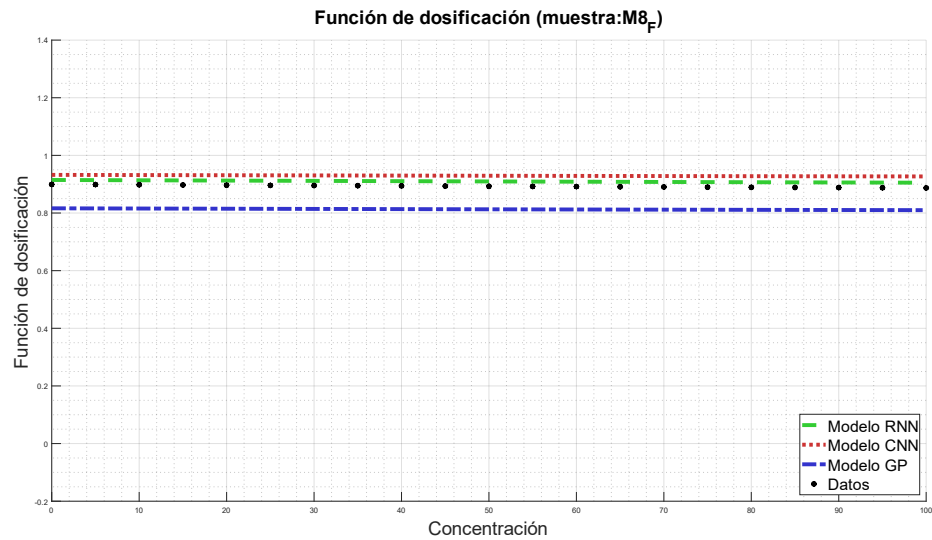


Figura 138: Simulación de la función de dosificación para la muestra R10.

Concentración crítica

En cuanto a la concentración crítica de esta muestra, los modelos que mejor reflejan el comportamiento de los datos son GP y RNN, mientras que el modelo CNN se aleja de la predicción de éstos. El error porcentual del modelo GP es de 6,26%, mientras que para RNN y CNN es de 0,47% y 2,60%, respectivamente. En la Figura 139 se observa el valor de la concentración crítica para la muestra R10.

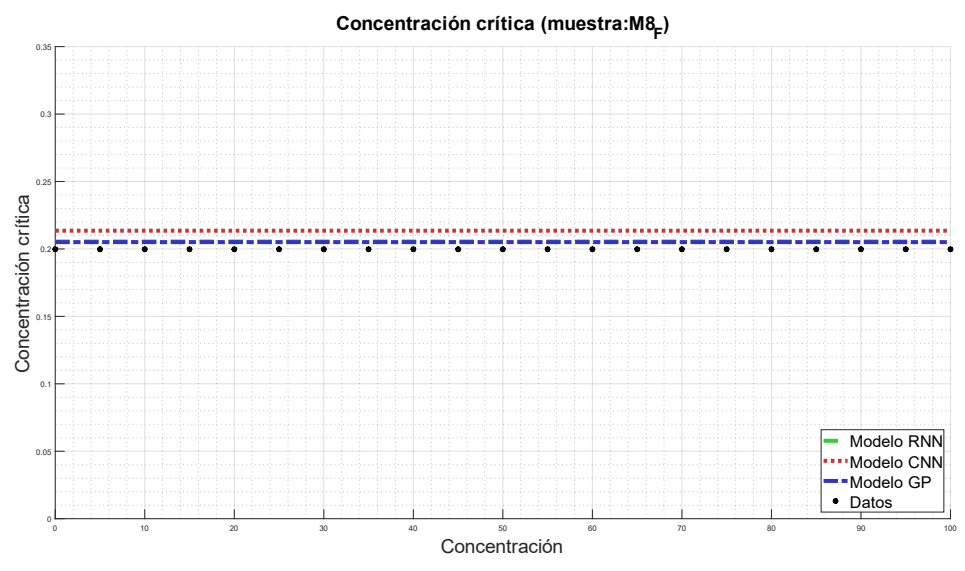


Figura 139: Simulación de la concentración crítica para la muestra R10.

B.11 Muestra R11

Función batch de Kynch

En la Figura 140 se observa la función batch de Kynch para la muestra R11, en función de la concentración. Para esta muestra, todos los modelos representan de buena manera el valor presentado por los datos experimentales. Para el coeficiente A se observa un error porcentual del 5,22% para el modelo GP, 2,24% para el modelo RNN y 1,74% para el modelo CNN, mientras que para el coeficiente b, se obtuvo un error porcentual para el modelo GP del 4,39%, para el modelo RNN un 0,02%, y 0,62% para el modelo CNN. Se puede observar para esta muestra que todos los modelos se ajustan correctamente, con rangos aceptables de error.

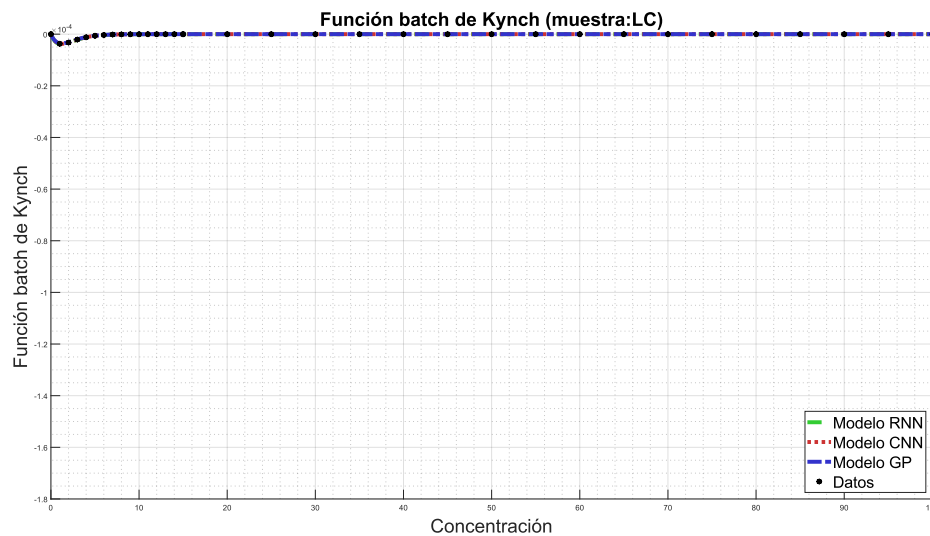


Figura 140: Simulación de la función batch de Kynch para la muestra R11.

Función de estrés efectivo del sólido

Para el caso de la muestra R11, en la Figura 141 se observa la curva descrita por la función de estrés efectivo del sólido. Se puede notar que para los 3 algoritmos se obtienen curvas que describen de buena manera a los datos

empíricos. Con respecto al coeficiente C , se obtuvo un error porcentual del 1,21% para GP, y 0,021% para los modelos RNN y CNN. Con esto, se tiene un error muy bajo para los 3 modelos estudiados. Por otro lado, el coeficiente d presenta un error porcentual del 7,87% para el modelo GP, 1,35% para el modelo RNN y 0,02% para el modelo CNN. Obteniéndose resultados similares para los 3 algoritmos, con respecto a lo observado en la práctica, como se observa en la Figura 141.

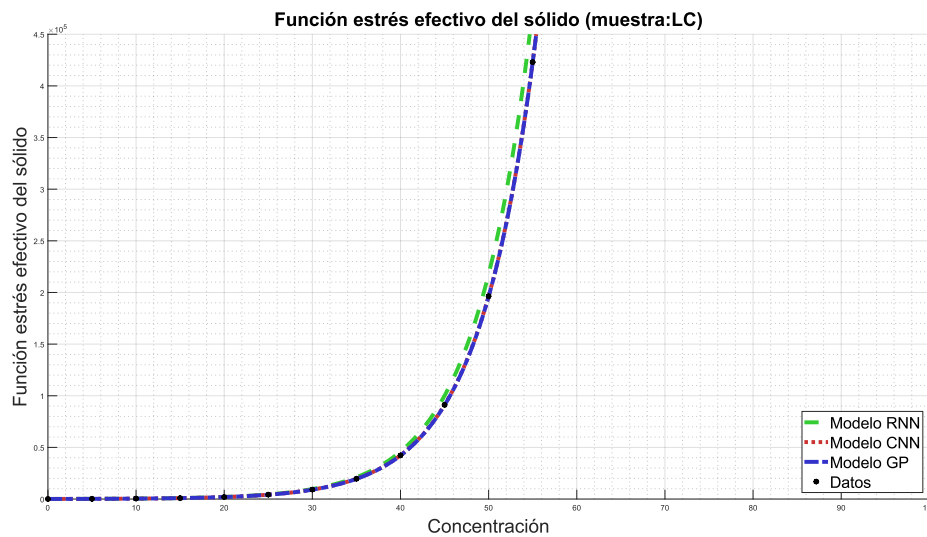


Figura 141: Simulación de la función estrés efectivo del sólido para la muestra R11.

Función de dosificación

En cuanto a la función de dosificación, en términos generales se observa en la Figura 142 que todos los modelos representan de buena manera la curva descrita por los datos experimentales. Para el coeficiente cuadrático p_1 se tiene un error porcentual del 3,78% para el modelo GP, 16,17% para el modelo RNN y 0,05% para el modelo CNN. Para el coeficiente lineal p_2 se observa un 0,03% de error para el modelo GP, 0,07% para el modelo RNN y 0,01% para el modelo CNN. Finalmente, para el coeficiente libre p_3 se tiene un error de 0,73% para el modelo GP, 0,93% para el modelo RNN, y 1,95% para el modelo CNN. Como análisis global de la función de

dosificación para esta muestra, se observa un ajuste aceptable para todos los modelos.

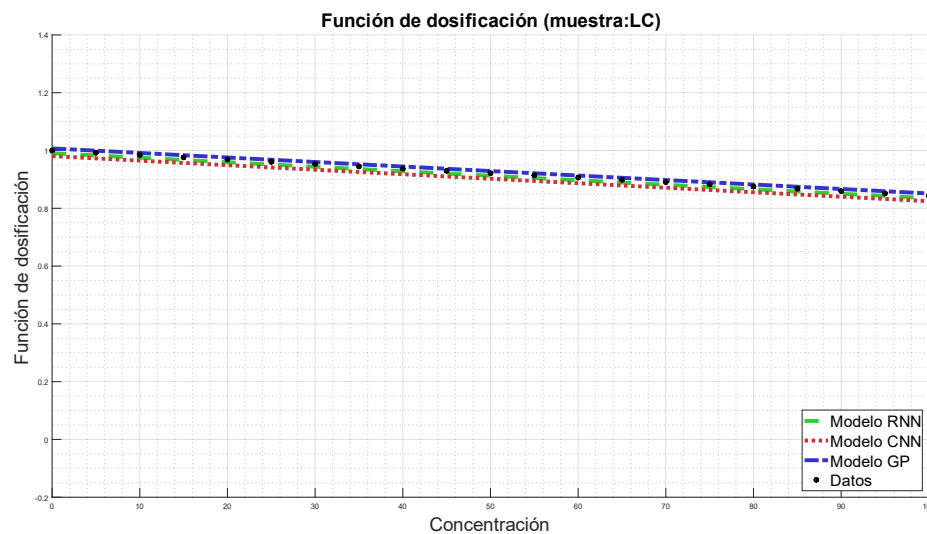


Figura 142: Simulación de la función de dosificación para la muestra R11.

Concentración crítica

En cuanto a la concentración crítica de esta muestra, los modelos que mejor reflejan el comportamiento de los datos son RNN y CNN, mientras que el modelo GP se aleja de la predicción de éstos. El error porcentual del modelo GP es de 6,31%, mientras que para RNN y CNN es de 0,47% y 2,60%, respectivamente. En la Figura 143 se observa el valor de la concentración crítica para la muestra R11.

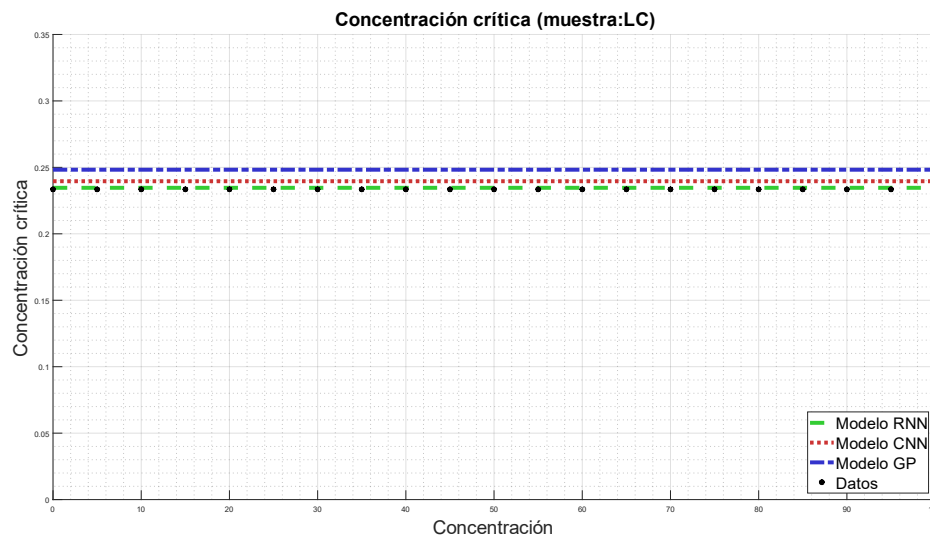


Figura 143: Simulación de la concentración crítica para la muestra R11.

B.12 Muestra R12

Función batch de Kynch

En la Figura 144 se observa la función batch de Kynch para la muestra R12, en función de la concentración. Para esta muestra, los modelos RNN y CNN vuelven a coincidir de mejor manera que el modelo GP, donde se distancia principalmente en la zona de máxima convexidad de la función. Para el coeficiente A se observa un error porcentual del 10,06% para el modelo GP, 0,015% para el modelo RNN y 0,2% para el modelo CNN, mientras que para el coeficiente b, se obtuvo un error porcentual para el modelo GP del 27,12%, para el modelo RNN un 2,13%, y 2,86% para el modelo CNN. Excluyendo el resultado obtenido mediante GP para el coeficiente A, todos los modelos se ajustan correctamente con la muestra, con rangos aceptables de error.

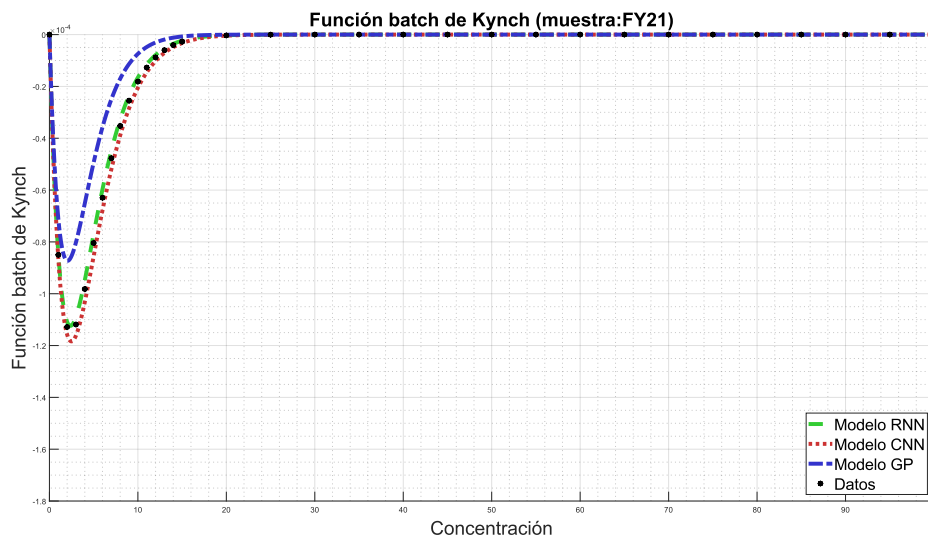


Figura 144: Simulación de la función batch de Kynch para la muestra R12.

Función de estrés efectivo del sólido

Para el caso de la muestra R12, en la Figura 145 se observa la curva descrita por la función de estrés efectivo del sólido. Se observa que para los 3 algoritmos se obtienen curvas que describen de buena manera a los datos empíricos. Con respecto al coeficiente C, se obtuvo un error porcentual del 0,61% para GP, 0,13% para el modelo RNN y 0,03% para el modelo CNN. Con esto, se tiene un error muy bajo para los 3 modelos estudiados. Por otro lado, el coeficiente d presenta un error porcentual del 4,99% para el modelo GP, 4,42% para el modelo RNN y 0,21% para el modelo CNN. Obteniéndose resultados similares para los 3 algoritmos, con respecto a lo observado en la práctica, como se observa en la Figura 145.

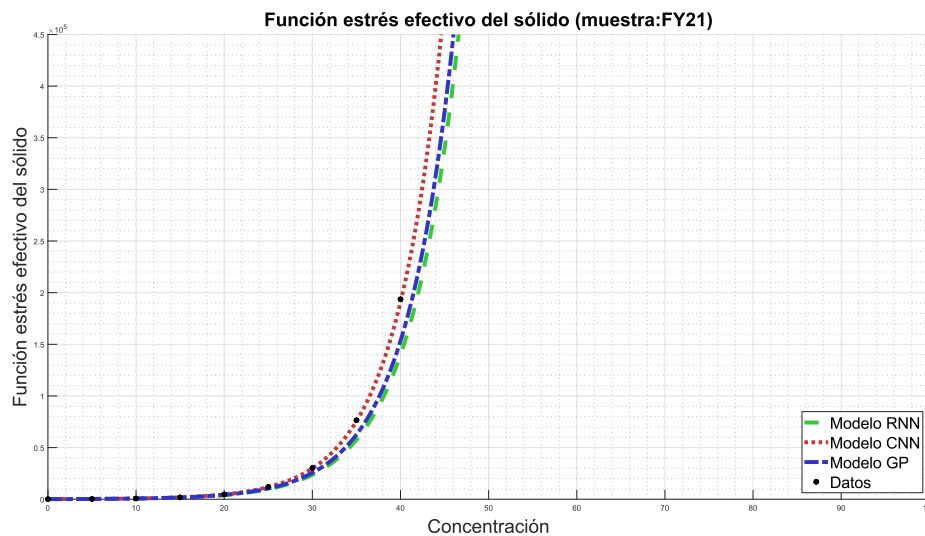


Figura 145: Simulación de la función estrés efectivo del sólido para la muestra R12.

Función de dosificación

En cuanto a la función de dosificación, en términos generales se observa en la Figura 146 que el modelo RNN se ajusta a la curva descrita por los datos, mientras que los modelos CNN y GP se distancian, especialmente éste último. Para el coeficiente cuadrático p_1 se tiene un error porcentual del 19,43% para el modelo GP, 4,19% para el modelo RNN y 16,88% para el modelo CNN. Para el coeficiente lineal p_2 se observa un 301,12% de error para el modelo GP, 138,20% para el modelo RNN y 968,54% para el modelo CNN. Finalmente, para el coeficiente libre p_3 se tiene un error de 0,73% para el modelo GP, 0,93% para el modelo RNN, y 1,95% para el modelo CNN. Como análisis global de la función de dosificación para esta muestra, se observa un ajuste aceptable para el modelo RNN, mientras que los demás modelos se alejan de la predicción deseada.

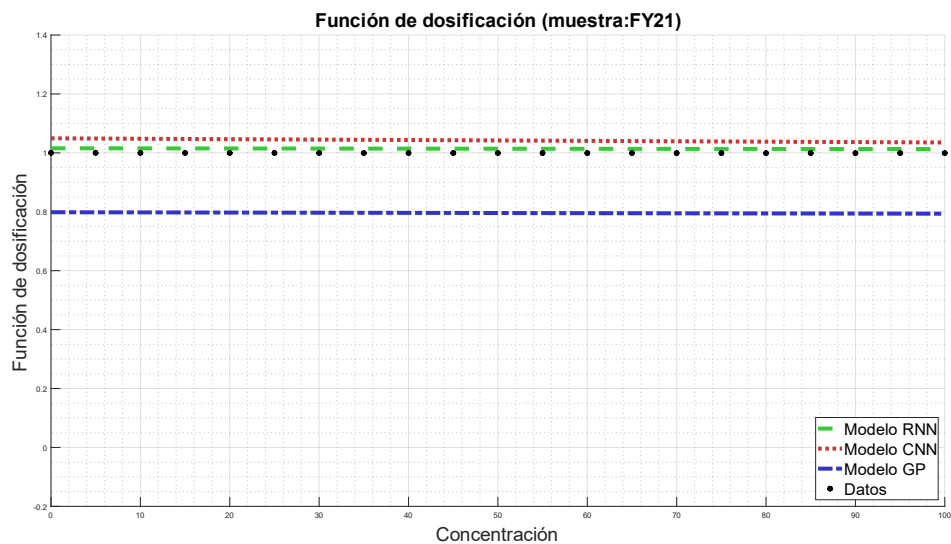


Figura 146: Simulación de la función de dosificación para la muestra R12.

Concentración crítica

En cuanto a la concentración crítica de esta muestra, los modelos que mejor reflejan el comportamiento de los datos son RNN y CNN, mientras que el modelo GP se aleja de la predicción de éstos. El error porcentual del modelo GP es de 25,62%, mientras que para RNN y CNN es de 0,025% y 2,27%, respectivamente. En la Figura 147 se observa el valor de la concentración crítica para la muestra R12.

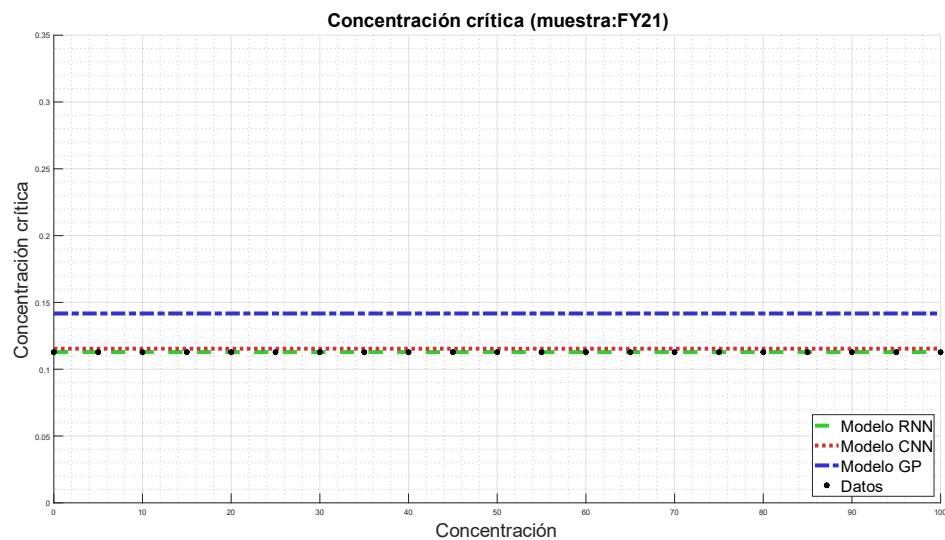


Figura 147: Simulación de la concentración crítica ϕ_c para la muestra R12.

B.13 Muestra R13

Función batch de Kynch

En la Figura 148 se observa la función batch de Kynch para la muestra R02, en función de la concentración. Para esta muestra, los modelos RNN y CNN vuelven a coincidir de mejor manera que el modelo GP, donde se distancia principalmente en la zona de máxima convexidad de la función, aunque ninguna curva llega a los datos experimentales. Para el coeficiente A se observa un error porcentual del 12,53% para el modelo GP, 0,50% para el modelo RNN y 0,07% para el modelo CNN, mientras que para el coeficiente b, se obtuvo un error porcentual para el modelo GP del 24,62%, para el modelo RNN un 11,14%, y 7,90% para el modelo CNN.

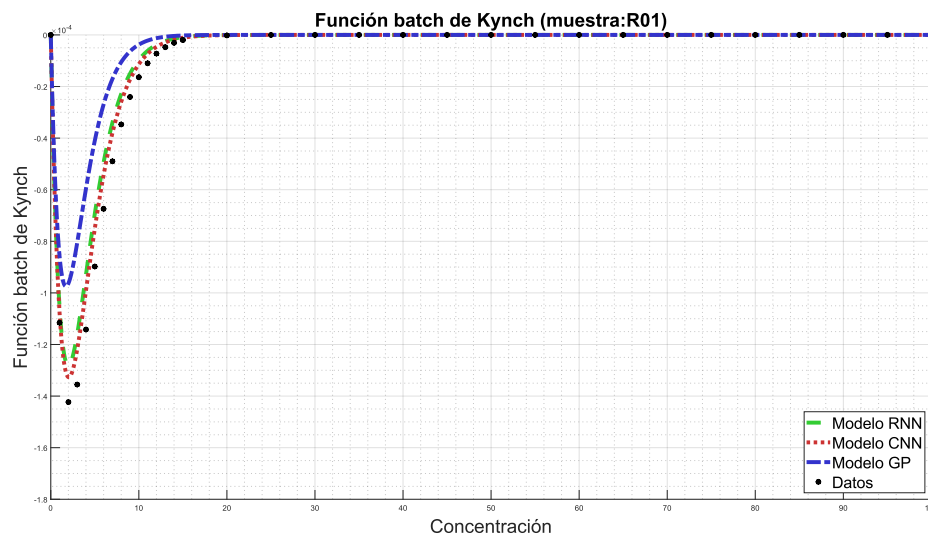


Figura 148: Simulación de la función batch de Kynch para la muestra R13.

Función de estrés efectivo del sólido

Para el caso de la muestra R13, en la Figura 149 se observa la curva descrita por la función de estrés efectivo del sólido. En la Figura se observa que para los modelos RNN y CNN se muestran curvas que describen de buena

manera a los datos empíricos, mientras que el modelo GP se alejan a medida que aumenta la concentración. Con respecto al coeficiente C, se obtuvo un error porcentual del 0,08% para GP, 0,16% para el modelo RNN y 0,05% para el modelo CNN. Con esto, se tiene un error muy bajo para los 3 modelos estudiados. Por otro lado, el coeficiente d presenta un error porcentual del 5,17% para el modelo GP, 0,06% para el modelo RNN y 0,95% para el modelo CNN. Obteniéndose resultados similares para los 3 algoritmos, con respecto a lo observado en la práctica, como se observa en la Figura 149.

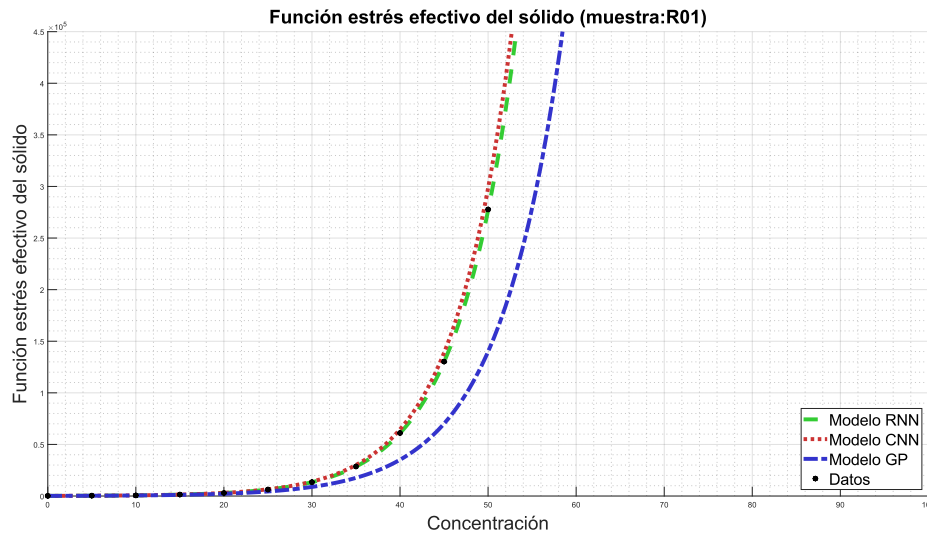


Figura 149: Simulación de la función estrés efectivo del sólido para la muestra R13.

Función de dosificación

En cuanto a la función de dosificación, en términos generales se observa en la Figura 150 que ningún modelo representa de buena manera el valor obtenidos por los datos experimentales, siendo curvas que se alejan, en especial los modelos CNN y GP. Para el coeficiente cuadrático p_1 se tiene un error porcentual del 496,95% para el modelo GP, 145,98% para el modelo RNN y 152,32% para el modelo CNN. Para el coeficiente lineal p_2 se observa un 59,65% de error para el modelo GP, 33,97% para el modelo RNN y 57,70% para el modelo CNN. Finalmente, para el coeficiente libre

p_3 se tiene un error de 41,26% para el modelo GP, 65,01% para el modelo RNN, y 30,26% para el modelo CNN. Como análisis global de la función de dosificación para esta muestra, no se representa el comportamiento de la muestra.

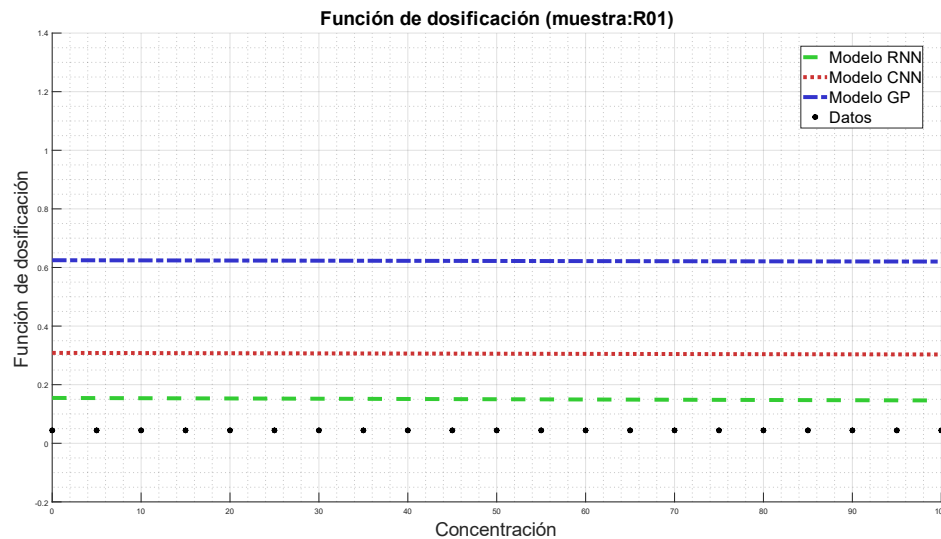


Figura 150: Simulación de la función de dosificación para la muestra R13.

Concentración crítica

En cuanto a la concentración crítica de esta muestra, todos los modelos representan de buena manera a los datos. El error porcentual del modelo GP es de 5,08%, mientras que para RNN y CNN es de 1,01% y 5,34%, respectivamente. En la Figura 151 se observa el valor de la concentración crítica para la muestra R13.

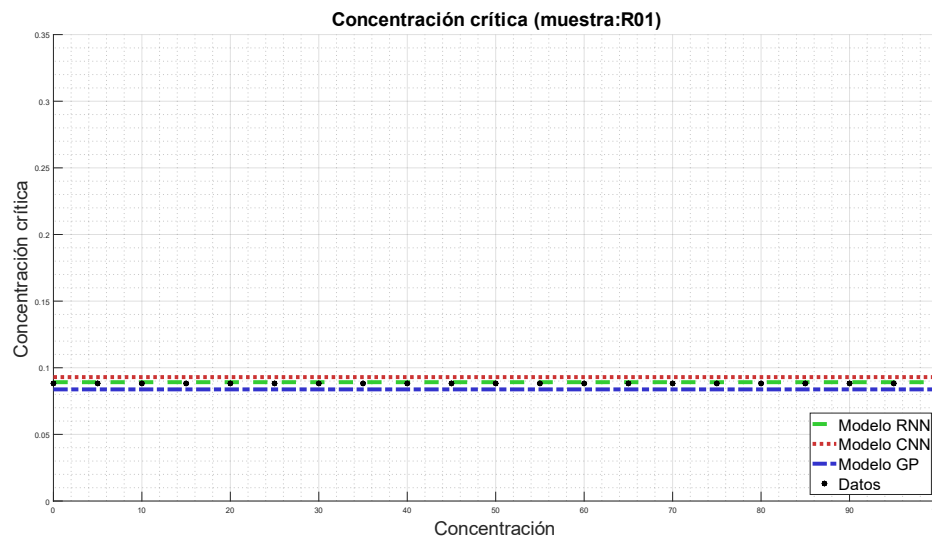


Figura 151: Simulación de la concentración crítica para la muestra R13.

B.14 Muestra R14

Función batch de Kynch

En la Figura 152 se observa la función batch de Kynch para la muestra R14, en función de la concentración. Para esta muestra, los modelos RNN coinciden de mejor manera que los demás modelos, donde se distancia principalmente en la zona de máxima convexidad de la función. Para el coeficiente A se observa un error porcentual del 26,76% para el modelo GP, y 1,74% para el modelo RNN y 21,96% CNN, mientras que para el coeficiente b, se obtuvo un error porcentual para el modelo GP del 3,60%, para el modelo RNN un 1,56%, y 7,50% para el modelo CNN. Excluyendo el resultado obtenido mediante GP y CNN para el coeficiente A, todos los modelos se ajustan correctamente con la muestra, con rangos aceptables de error.

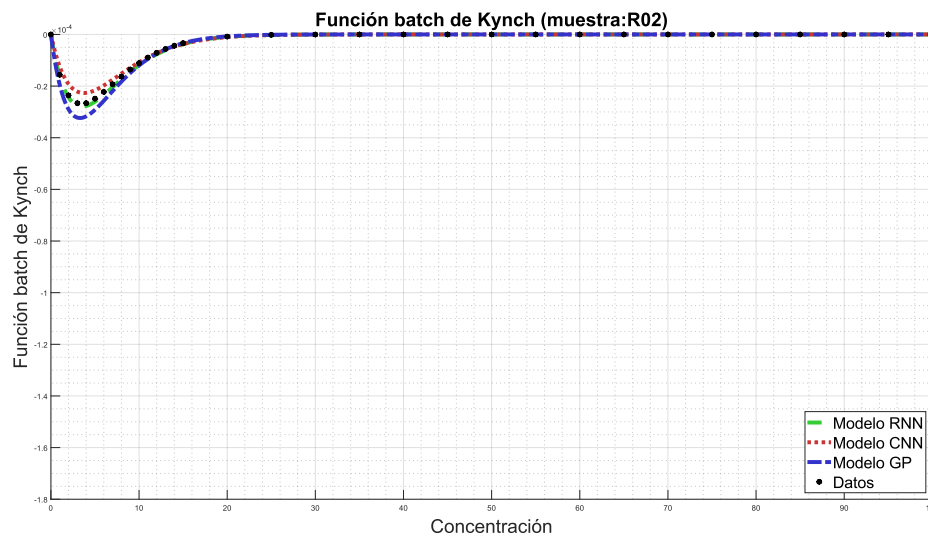


Figura 152: Simulación de la función batch de Kynch para la muestra R14.

Función de estrés efectivo del sólido

Para el caso de la muestra R01, en la Figura 153 se observa la curva descrita por la función de estrés efectivo del sólido. Se observa que el modelo CNN coincide con el valor mostrado por los datos, mientras que el modelo RNN se aleja y el modelo GP aún más. Con respecto al coeficiente C, se obtuvo un error porcentual del 5,30% para GP, 0,03% para el modelo RNN y 0,02% para el modelo CNN. Con esto, se tiene un error muy bajo para los 3 modelos estudiados. Por otro lado, el coeficiente d presenta un error porcentual del 10,83% para el modelo GP, 4,45% para el modelo RNN y 0,01% para el modelo CNN. Obteniéndose resultados similares para los 3 algoritmos, con respecto a lo observado en la práctica, como se observa en la Figura 153.

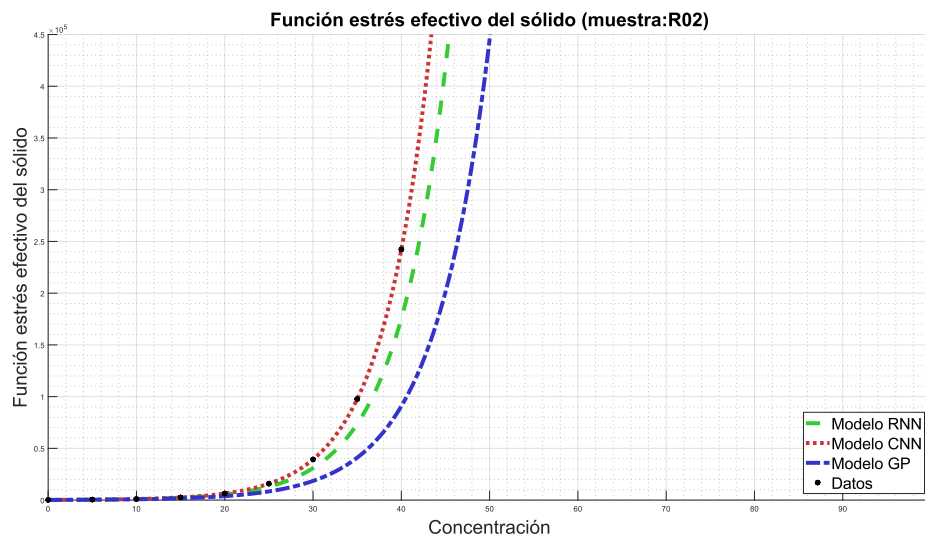


Figura 153: Simulación de la función estrés efectivo del sólido para la muestra R14.

Función de dosificación

En cuanto a la función de dosificación, en términos generales se observa en la Figura 154 que ningún modelo representa de buena manera el valor obtenidos por los datos experimentales, siendo curvas que se alejan, en especial los modelos CNN y GP. Para el coeficiente cuadrático p_1 se tiene un error porcentual del 16,54% para el modelo GP, 22,67% para el modelo RNN y 20,05% para el modelo CNN. Para el coeficiente lineal p_2 se observa un 73,75% de error para el modelo GP, 0,52% para el modelo RNN y 14,69% para el modelo CNN. Finalmente, para el coeficiente libre p_3 se tiene un error de 218,44% para el modelo GP, 8,44% para el modelo RNN, y 94,05% para el modelo CNN. Como análisis global de la función de dosificación para esta muestra, no se representa el comportamiento de la muestra.

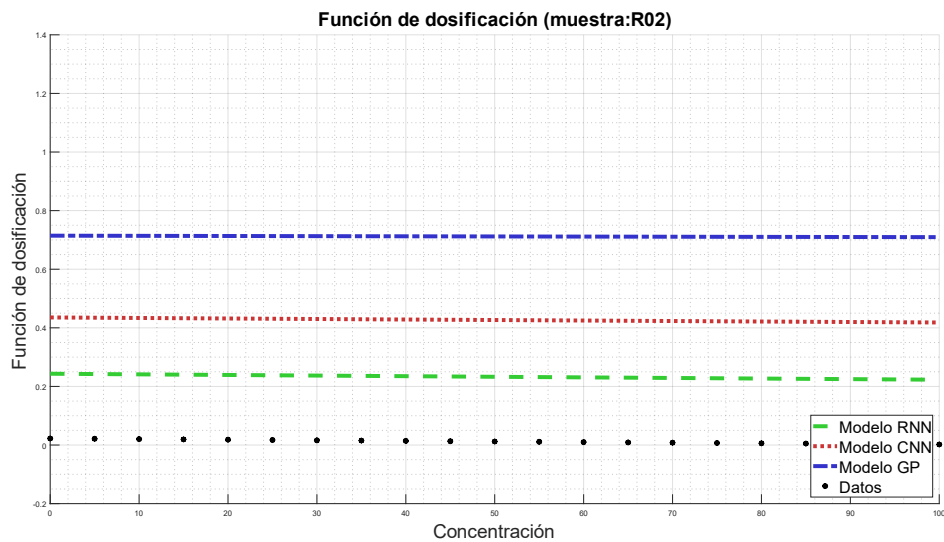


Figura 154: Simulación de la función de dosificación para la muestra R14.

Concentración crítica

En cuanto a la concentración crítica de esta muestra, los modelos que mejor reflejan el comportamiento de los datos son RNN y CNN, mientras que el modelo GP se aleja de la predicción de éstos. El error porcentual del modelo GP es de 59,57%, mientras que para RNN y CNN es de 5,02% y 10,53%, respectivamente. En la Figura 155 se observa el valor de la concentración crítica para la muestra R14.

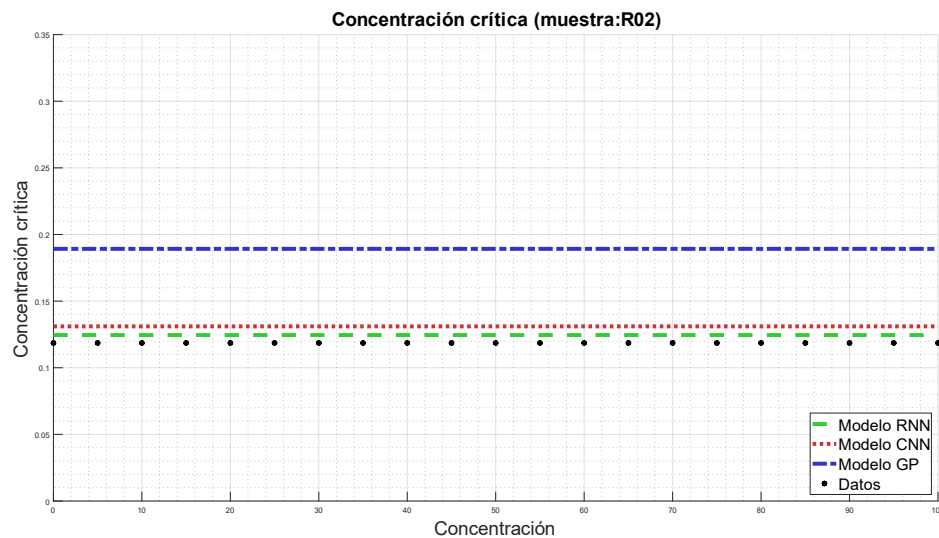


Figura 155: Simulación de la concentración crítica para la muestra R14.

B.15 Muestra R15

Función batch de Kynch

En la Figura 156 se observa la función batch de Kynch para la muestra R15, en función de la concentración. Para esta muestra, los modelos RNN y CNN vuelven a coincidir de mejor manera que el modelo GP, donde se distancia principalmente en la zona de máxima convexidad de la función. Para el coeficiente A se observa un error porcentual del 17,54% para el modelo GP, 33,82% para el modelo RNN y 49,24% CNN, mientras que para el coeficiente b, se obtuvo un error porcentual para el modelo GP del 3,60%, para el modelo RNN un 1,56%, y 7,50% para el modelo CNN.

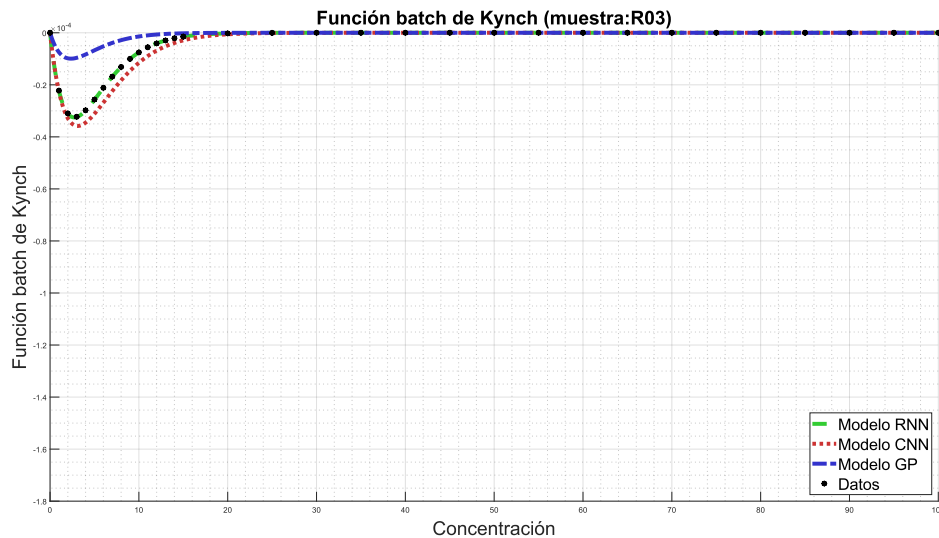


Figura 156: Simulación de la función batch de Kynch para la muestra R15.

Función de estrés efectivo del sólido

Para el caso de la muestra R01, en la Figura 157 se observa la curva descrita por la función de estrés efectivo del sólido. En la Figura se observa que para los 3 algoritmos se obtienen curvas que describen de buena manera a los datos empíricos. Con respecto al coeficiente C, se obtuvo un error porcentual del 3,89% para GP, 0,012% para el modelo RNN y 2,9% para el modelo CNN. Con esto, se tiene un error muy bajo para los 3 modelos estudiados. Por otro lado, el coeficiente d presenta un error porcentual del 9,50 para el modelo GP, 4,0% para el modelo RNN y 0,04% para el modelo CNN. Obteniéndose resultados similares para los 3 algoritmos, con respecto a lo observado en la práctica, como se observa en la Figura 157.

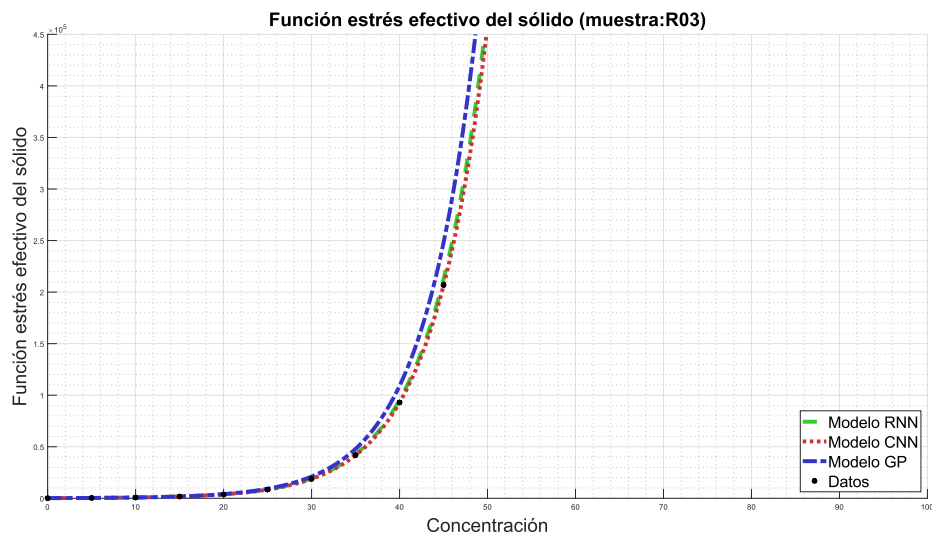


Figura 157: Simulación de la función estrés efectivo del sólido para la muestra R15.

Función de dosificación

En cuanto a la función de dosificación, en términos generales se observa en la Figura 158 que el modelo RNN se ajusta a la curva descrita por los datos, mientras que los modelos GP y CNN se distancian, manteniendo esa distancia a medida que aumenta la concentración. Para el coeficiente cuadrático p_1 se tiene un error porcentual del 18,97% para el modelo GP, 75,72% para el modelo RNN y 2,06% para el modelo CNN. Para el coeficiente lineal p_2 se observa un 39,41% de error para el modelo GP, 16,62% para el modelo RNN y 14,48% para el modelo CNN. Finalmente, para el coeficiente libre p_3 se tiene un error de 674,24% para el modelo GP, 30,99% para el modelo RNN, y 287,74% para el modelo CNN. Como análisis global de la función de dosificación para esta muestra, se observa un ajuste aceptable para el modelo RNN, mientras que los demás modelos se alejan de la predicción deseada.

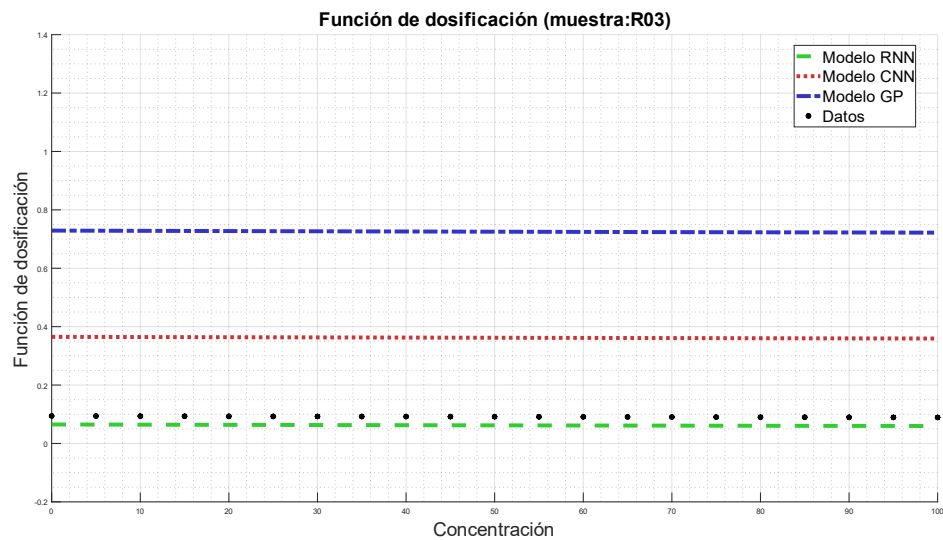


Figura 158: Simulación de la función de dosificación para la muestra R15.

Concentración crítica

En cuanto a la concentración crítica de esta muestra, el modelo que mejor refleja el comportamiento de los datos es el modelo RNN, mientras que los modelos CNN y GP se aleja de la predicción de éstos, en especial éste último. El error porcentual del modelo GP es de 22,69%, mientras que para RNN y CNN es de 1,68% y 7,24%, respectivamente. En la Figura 159 se observa el valor de la concentración crítica para la muestra R15.

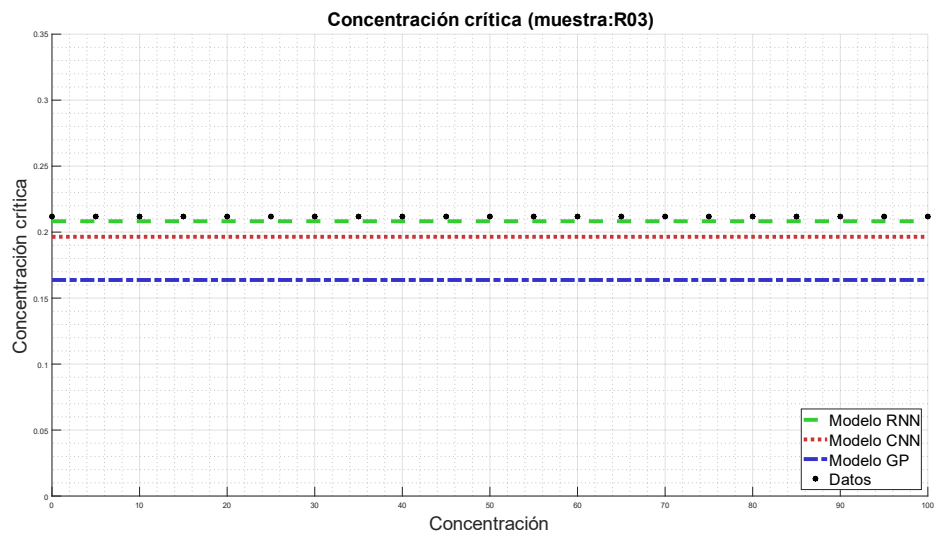


Figura 159: Simulación de la concentración crítica para la muestra R15.

