



UNIVERSIDAD DE CHILE

UNIVERSIDAD DE CHILE

EL7012

CONTROL INTELIGENTE DE SISTEMAS, OTOÑO

Ejercicio N°1

Autor:

Elias Obrequé
Gustavo Ceballo
Maibeth Sánchez

31 de mayo de 2020

Índice

1. Introducción	1
1.1. Problema 1	1
2. Generación de Datos	3
3. Modelos de predicción	6
3.1. Modelo lineal	6
3.1.1. Predicciones a 1, 8, y 16 pasos	8
3.1.2. Intervalos de predicción	9
3.2. Modelo difuso Takagi-Sugeno Tipo-1	13
3.2.1. Predicciones a 1, 8, y 16 pasos	18
3.2.2. Intervalos de predicción	18
3.3. Modelo de red neuronal	25
3.3.1. Predicciones a 1, 8, y 16 pasos	34
3.3.2. Intervalo de predicción	40
4. Resultados previos y conclusión	43
5. Intervalo de predicción con Joint Supervision	44
6. Problema 2	48
6.1. Selección de Datos	48
6.2. Modelo de red neuronal	49

Índice de figuras

1. Comportamiento de la planta con una entrada de escalón unitario. . .	4
2. Señal APRBS con Amplitud entre -1 y +1.	4
3. Respuesta de la serie no lineal.	5
4. Predicción del modelo Lineal.	7
5. Predicción del modelo Lineal.	8
6. Curvas para Intervalo del modelo lineal considerando predicción a 1 paso y $\alpha = 0,8$	10
7. Curvas para Intervalo del modelo lineal considerando predicción a 1 paso y $\alpha = 1,5$	10
8. Curvas para Intervalo del modelo lineal	11
9. Índice de Sensibilidades.	14
10. Índice de Sensibilidades.	15
11. Clusters del Modelo Difuso Tipo 1.	16

12.	Respuesta del Modelo Difuso Tipo 1.	19
13.	Intervalo difuso. Método de la Covarianza.	22
14.	Intervalo difuso. Método MinMax	24
15.	Estructura de la red neuronal perceptrón	25
16.	RMSE para diferente número de neuronas.	28
17.	Sensibilidad para un número de neuronas entre $[2 - 21]$ en la capa oculta.	28
18.	Sensibilidad para un número de neuronas entre $[22 - 41]$ en la capa oculta.	29
19.	Índice de sensibilidad para 4 y 3 entradas.	30
20.	Comparación entre la salida de la red neuronal y el valor real del conjunto de validación.	31
21.	Histograma del error de cada conjunto de datos.	31
22.	RMSE para diferente número de neuronas.	32
23.	Sensibilidad para un número de neuronas entre $[2 - 21]$ en la capa oculta con 3 entradas.	32
24.	Sensibilidad para un número de neuronas entre $[22 - 41]$ en la capa oculta con 3 entradas.	33
25.	Resultados del modelo neuronal con 10 neuronas en la capa oculta en comparación con los datos reales.	34
26.	Resultados de entrenamiento mostrando la regresión de cada conjunto.	34
27.	Predicción a 1 paso con modelo neuronal.	35
28.	Modelo de predicción para j pasos del modelo neuronal.	36
29.	Predicción a 8 paso con modelo neuronal.	37
30.	Predicción a 16 paso con modelo neuronal.	39
31.	Uso del método de covarianza para estimar el intervalo de predicción a 1, 8 y 16 con $\alpha = 5$	41
32.	Uso del método de covarianza para estimar el intervalo de predicción a 1, 8 y 16 con $\alpha = 10$	42
33.	Metodología para desarrollar intervalos de predicción	44
34.	Uso del método de covarianza para estimar el intervalo de predicción a 1, 8 y 16 con $\alpha = 10$	45

Índice de tablas

1.	Errores o Métricas de bondad de ajuste a 1 paso	7
2.	Métricas de intervalo para diferentes valores de α	9
3.	Métricas de porcentaje de cobertura y ancho del intervalo para diferentes valores de $\alpha = 0,8$ considerando intervalos de predicción a 1, 8 y 16 pasos respectivamente.	11

4.	Índices de Error para el Análisis de Sensibilidades.	15
5.	Índices de Error para el Modelo Difuso	18
6.	Desempeño del Modelo difuso a j-pasos	19
7.	Intervalos de predicción. Método de la Covarianza	21
8.	Intervalos de predicción. Método MinMax	23
9.	Experimento para ver la influencia del sobre ajuste.	27
10.	Valores MSE para los 3 conjuntos de datos evaluados en una red con 4 entradas y otra con 3 entradas.	29
11.	Métricas finales del modelo neuronal	33
12.	Métricas de evaluación para 1, 8 y 16 pasos adelante.	38
13.	Métricas de evaluación de intervalo para diferentes pasos con $\alpha = 5$. .	41
14.	Métricas de evaluación de intervalo para diferentes pasos con $\alpha = 10$. .	42
15.	Comparación de los modelos propuestos	43
16.	Comparación de modelos de intervalo.	43
17.	Métricas de evaluación de intervalo para diferentes pasos con $\alpha = 10$. .	46

Ejercicio N°1

1. Introducción

La mayoría de los sistemas tienen un comportamiento no lineal, excepto en un determinado rango de operación donde pueden ser considerados lineales. En ocasiones un modelo lineal es insuficiente para explicar un fenómeno por lo que se debe recurrir a Modelos No Lineales.

Los modelos basados en redes neuronales y sistemas difusos han sido empleados para la identificación, dado que han mostrado una grancapacidad para aproximar funciones no lineales desconocidas, además de que ofrecen una estructura general tan compleja o sencilla como el problema lo demande. Por un lado, las redes neuronales son estructuras matemáticas inspiradas en las estructuras biológicas compuestas por neuronas, en donde a través de elementos relativamente simples (llamados neuronas) que efectúan operaciones y procesos muy sencillos logran, mediante interconexiones, realizar procesos complejos y procesamiento de información en paralelo. En general se consideran redes neuronales conformadas por “capas” de neuronas, donde una capa procesa la información recibida en paralelo y la envía a la siguiente capa de neuronas. Por otro lado, los sistemas difusos son también estructuras matemáticas, aunque estos se encuentran inspirados en los procesos de pensamiento y toma de decisiones llevados a cabo por los humanos, así como en la teoría de conjuntos difusos propuesta por Lofti Zadeh.

En este ejercicio se dará solución al Problema 1 con ayuda del material del curso [1][2]. Los códigos utilizados para los modelos se encuentran en [3].

1.1. Problema 1

Considere la siguiente serie no lineal dinámica:

$$\begin{aligned} y(k) = & (0,8 - 0,5\exp\{-y^2(k-1)\})y(k-1) \\ & - (0,3 + 0,9\exp\{-y^2(k-1)\})y(k-2) \\ & + u(k-1) + 0,2u(k-2) + 0,1u(k-1)u(k-2) + e(k) \end{aligned} \quad (1)$$

donde el ruido del sistema

$$e(k) = 0,5\exp\{-y^2(k-1)\}\beta(k) \quad (2)$$

depende del estado previo de la salida del modelo, y $\beta(k)$ es un ruido blanco.

Como usted sabe existen varias técnicas que se pueden emplear para la modelación a partir de estos datos, por lo que debe seleccionar el tipo de modelo más adecuado para este tipo de sistema. Para este trabajo se le pide detallar la metodología utilizada para:

- a) Generar 600 datos a partir de esta serie. Considere 55 % para entrenamiento, 25 % test y 20 % validación.
- b) Obtener un modelo de predicción lineal, difuso tipo-1 (T&S) y neuronal para la salida. Evaluar las predicciones a 1, 8 y 16 pasos. Comparar el desempeño de todos los modelos a partir de las métricas más apropiadas tales como RMSE, MAPE, MAE, entre otras. Comente.
- c) Construir el intervalo de predicción de los modelos obtenidos en b) utilizando el método de la covarianza.
- d) Evaluar los intervalos de predicción obtenidos en b) realizando predicciones a 1, 8, y 16 pasos. Comparar el desempeño de los modelos a partir de las métricas más apropiadas tales como ancho del intervalo, probabilidad de cobertura, entre otras.
- e) Construir el intervalo de predicción del modelo difuso encontrado en a) con el método de optimización min-max. Compare este intervalo de predicción con el intervalo obtenido utilizando el método de la covarianza. Comente.
- f) Construir el intervalo de predicción neural utilizando el método de Joint Supervision. Compare con los métodos anteriores.
- g) Seleccione el modelo más apropiado y justifique.

2. Generación de Datos

En esta etapa es necesario generar datos que representen la dinámica del sistema en la mayor cantidad de rangos de operación posibles, ya que el modelo obtenido tiene un ancho de banda acotado, y por lo tanto las dinámicas definidas por fuera de dicha banda podrían no ser representadas adecuadamente. Para lo cual se debe diseñar una entrada $u(k)$ que excite a la planta en el rango de frecuencias en que se encuentran los fenómenos de interés.

En este trabajo se propone el uso de señales binarias pseudo aleatorias (Pseudo Random Binary Signal, PRBS), ya que es una de las señales más utilizadas en identificación de sistemas. Esta es una señal periódica, determinística y que posee principalmente propiedades similares al ruido blanco (contenido muy rico en frecuencias) [4].

Para generar la señal se suponen los siguientes parámetros de interés $f_{min} = 0,2Hz$, $f_{max} = 1Hz$ y tiempo de muestreo $T_S = 0,01$. Con los parámetros anteriores, y utilizando la expresión

$$n = \frac{\log(f_c/f_{min} + 1)}{\log(2)} \quad (3)$$

con $f_c = 2,5 * f_{max} = 2,5Hz$, se genera una PRBS de orden $n = 4$, por lo que el largo máximo corresponde a $N = 2^n - 1 = 15$. A su vez, la cantidad de muestras por bit son $N_s = 40$, lo cual supera el número de iteraciones en el que el sistema se establece ante una entrada escalón sin ruido, aproximadamente 20 iteraciones según la Figura 1. Por lo que la PRBS debe ser replicada 400 veces con diferentes condiciones iniciales para obtener los 6000 datos de interés. Finalmente se genera la APRBS variando la amplitud aleatoriamente de la PRBS generada, Fig.2 y se aplica a la serie no lineal como se muestra en la Fig.3.

Una vez obtenidos los datos experimentales de entrada-salida, éstos son clasificados en tres conjuntos con distinta información: datos de entrenamiento, datos de validación y datos de prueba; esto con el fin de evaluar adecuadamente los modelos generados. El conjunto de entrenamiento se utiliza para determinar los parámetros del modelo. El conjunto de prueba permite comparar distintas estructuras de los modelos generados. Finalmente, el conjunto de validación permite verificar el sobreajuste del modelo óptimo obtenido, evaluándolo en un nuevo conjunto de datos (distintos a los datos del conjunto de entrenamiento y prueba), analizando su capacidad de generalización. En este caso se utiliza una división de 55 % de los datos para entrenamiento, 25 % para prueba y 20 % validación.

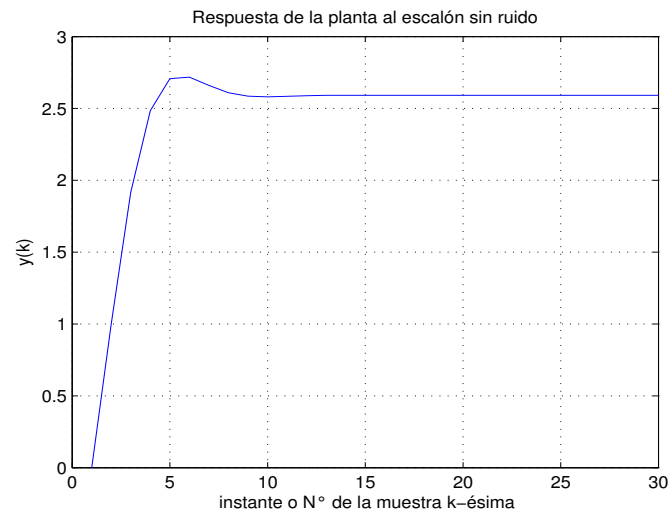


Figura 1: Comportamiento de la planta con una entrada de escalón unitario.

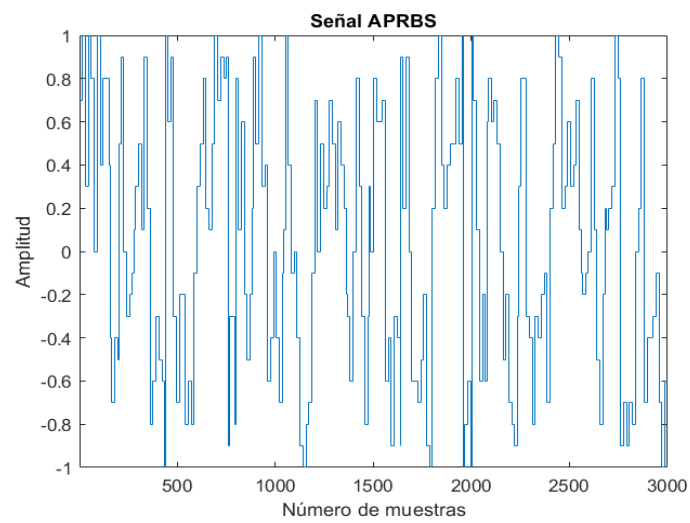


Figura 2: Señal APRBS con Amplitud entre -1 y +1.

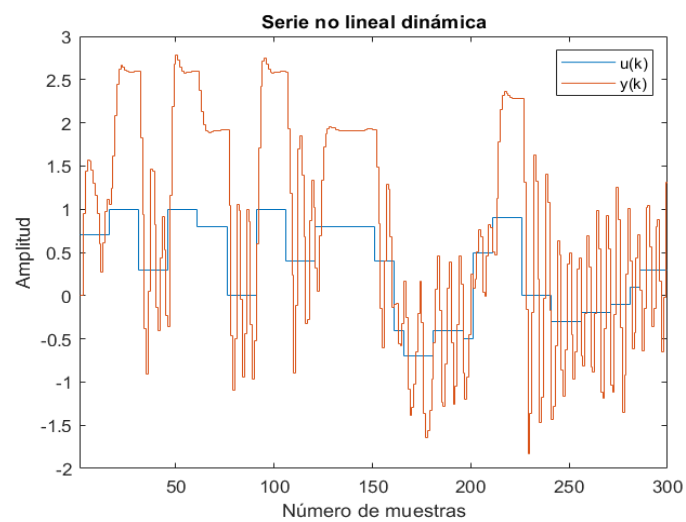


Figura 3: Respuesta de la serie no lineal.

3. Modelos de predicción

3.1. Modelo lineal

En este caso, supondremos que se ajustará un modelo lineal suponiendo que el sistema real es lineal con ruido blanco gaussiano aditivo, es decir,

$$y(k) = a_1y(k-1) + a_2y(k-2) + b_1u(k-1) + b_2u(k-2) + e(k) \quad (4)$$

Luego, se propone un modelo lineal para llevar a cabo la predicción a 1 paso, de modo tal que: Predicción a 1 paso:

$$\hat{y}(k) = \hat{a}_1y(k-1) + \hat{a}_2y(k-2) + \hat{b}_1u(k-1) + \hat{b}_2u(k-2) \quad (5)$$

Este modelo no considera un valor constante o bias dado el supuesto que el sistema es lineal con ruido blanco aditivo. En caso que se sospechara que existe un bias o tendencia (trend) en el sistema, se puede agregar otro vector de unos a la matriz de regresores (o matriz de información). Para llevar a cabo la estimación de los parámetros del modelo se utilizó la técnica de mínimos cuadrados, es decir:

$$\hat{\theta} = (Xent^T * Xent)^{-1} * Xent^T * \hat{y}(k) \quad (6)$$

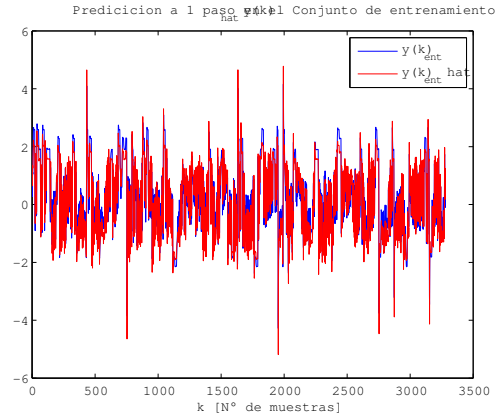
En que $\hat{\theta} = [\hat{a}_1 \quad \hat{a}_2 \quad \hat{b}_1 \quad \hat{b}_2]^T$ es el vector de parámetros y $Xent$ es la matriz de regresores con los valores de las n muestras ordenados por filas.

Los valores que se obtuvieron de los parámetros fueron los siguientes:

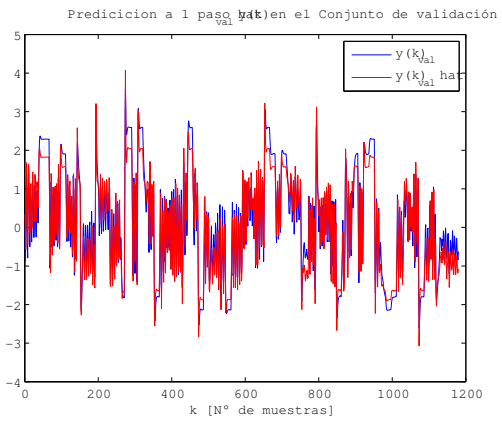
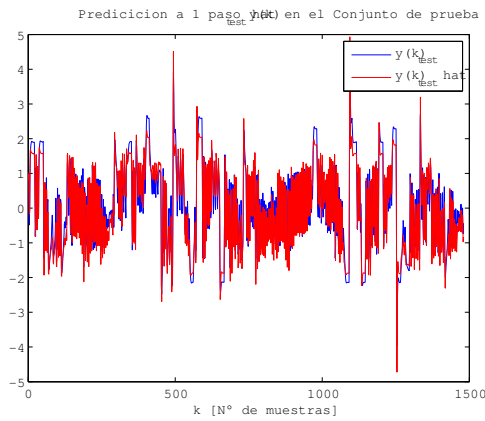
$$\hat{\theta} = \begin{pmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{b}_1 \\ \hat{b}_2 \end{pmatrix} = \begin{pmatrix} 0,8601 \\ -0,6930 \\ 0,9724 \\ 0,3486 \end{pmatrix} \quad (7)$$

En la Fig. 4, se muestran las gráficas de la predicción a un paso en cada conjunto de datos, a saber, conjunto de entrenamiento, prueba y validación. Esto es solo a modo de comparación puesto que finalmente, lo relevante de nuestro modelo, es el comportamiento o bondad de ajuste en el conjunto de validación. Evidentemente que los datos de los parámetros usados en las predicciones, para todos los conjuntos, son aquellos obtenidos con el conjunto de entrenamiento, sin embargo, la matriz de reporesores y salidas cambian dependiendo del conjunto que se esté usando para predicción. A continuación, en la Tabla 1, se presentan las métricas de bondad del

Ejercicio N°1



(a) Predicción a 1 paso en conjunto de entrenamiento



(b) Predicción a 1 paso en conjunto de prueba (c) Predicción a 1 paso en conjunto de validación

Figura 4: Predicción del modelo Lineal.

ajuste o errores en los diversos conjuntos de datos, a saber, conjunto de datos de entrenamiento, prueba o test y validación.

Tabla 1: Errores o Métricas de bondad de ajuste a 1 paso

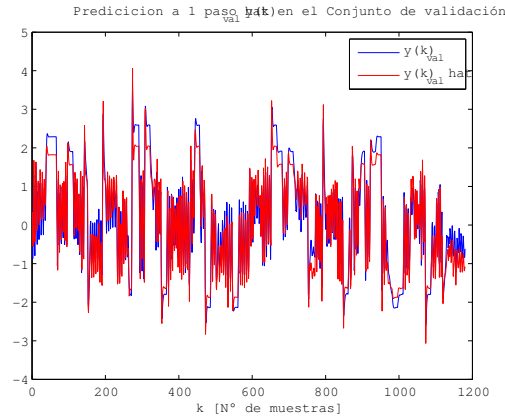
Métricas	Conjunto Entrenamiento	Conjunto prueba	Conjunto validación
RMSE	0.0869	0.0559	0.0954
MAPE	162.2793	220.0863	238.2150
MAE	0.3569	0.3465	0.6020

Es interesante notar que los ajustes del modelo, en todos los conjuntos, son bastante comparables entre sí, es decir, en general, los errores son similares en cuanto a RMSE

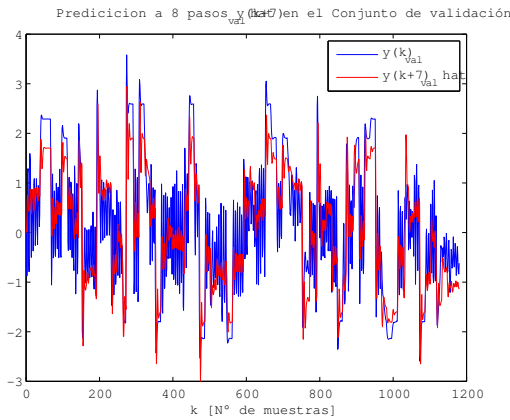
y MAE. Esto también se puede apreciar al observar la Fig. 4 respectivamente. Si bien, el conjunto de entrenamiento presenta el menor MAPE, a su vez, el conjunto de entrenamiento, presenta el menor RMSE.

3.1.1. Predicciones a 1, 8, y 16 pasos

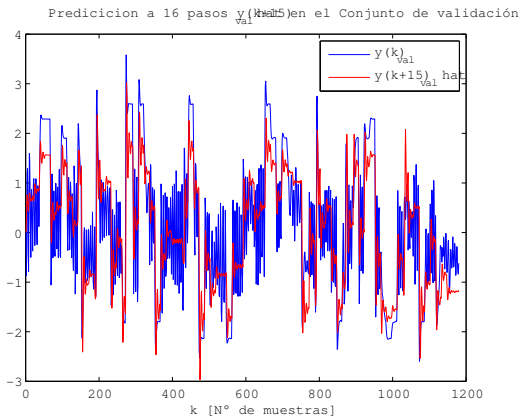
Para efectos de apreciar la robustez de este modelo, llevaremos a cabo la predicción a 8 y 16 pasos respectivamente, en el conjunto de validación del modelo lineal, para luego compararlos con los demás modelos, a saber, modelo de Takagi y Sugeno y Modelo Neuronal para el mismo conjunto de validación, Fig. 5.



(a) Predicción a 1 paso en el conjunto de validación para el Modelo Lineal



(b) Predicción a 8 pasos en el conjunto de validación para el Modelo Lineal.



(c) Predicción a 16 pasos en el conjunto de validación para el Modelo Lineal.

Figura 5: Predicción del modelo Lineal.

3.1.2. Intervalos de predicción

En este caso, se pueden definir 2 modelos adicionales que den cuenta de un intervalo que contenga un porcentaje de las muestras al interior del mismo. A saber, podríamos determinar un modelo por arriba (y_u) y uno por abajo (y_l) del modelo de valor esperado y_{val} , con los datos del conjunto de validación, determinado en la sección anterior.

Estos modelos se podrían construir considerando la adición de la varianza de las salidas del conjunto de validación entre el valor real y el estimado, es decir, el intervalo tendría la siguiente expresión para ambos modelos:

$$\hat{y}_u(k) = X_{val}(k)\hat{\theta}(k) + \alpha[Var(y(k)_{val} - \hat{y}_{val}(k))]^{1/2} \quad (8)$$

$$\hat{y}_l(k) = X_{val}(k)\hat{\theta}(k) - \alpha[Var(y(k)_{val} - \hat{y}_{val}(k))]^{1/2} \quad (9)$$

En que α es un parámetro de ajuste que tiene relación con el porcentaje de cobertura que tendría el intervalo, $Var()$ es el operador varianza de las salidas reales y estimadas en el conjunto de validación y $X_{val}(k)\hat{\theta}(k)$ es el modelo obtenido de valor esperado visto en la sección anterior, es decir,

$$\hat{y}_{val}(k) = X_{val}(k)\hat{\theta}(k) \quad (10)$$

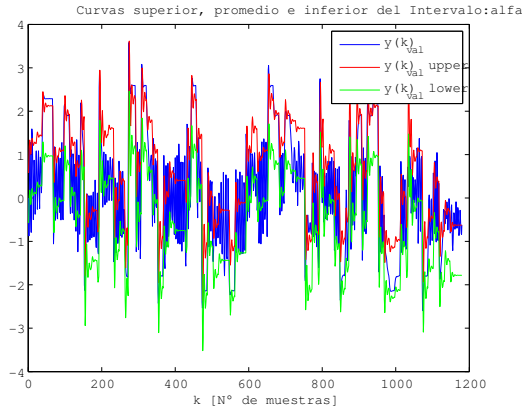
Así entonces, podemos determinar los modelos por arriba y por abajo del modelo clásico de valor esperado, los cuales nos definen el intervalo. En la Fig. 6, se muestran las gráficas de las 3 curvas o modelos para la predicción a un paso con $\alpha = 0,8$.

Ahora bien, podemos ir ajustando el valor de α para aumentar o disminuir el porcentaje de cobertura de las muestras de validación. En la Tabla 2, se muestran las métricas PICP y PINAW las cuales miden el porcentaje de cobertura y ancho del intervalo respectivamente, para diferentes valores de α de la predicción a 1 paso. A modo de ejemplo, en la Fig. 7 se muestra el caso en que el parámetro α se aumenta a 1.5.

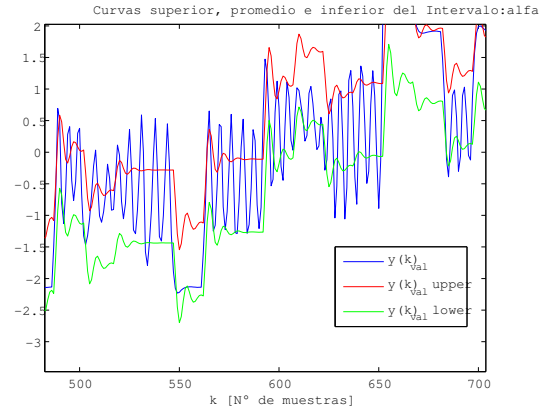
Tabla 2: Métricas de intervalo para diferentes valores de α .

Parámetro α	0,5	0,8	0,9	1	1,5	2
PICP (%)	32.2034	52.5424	58.7288	65.3390	885,593	96.1017
PINAW	11.687	18.6992	21.0366	23.7339	35.0609	46.7479

Ejercicio N°1

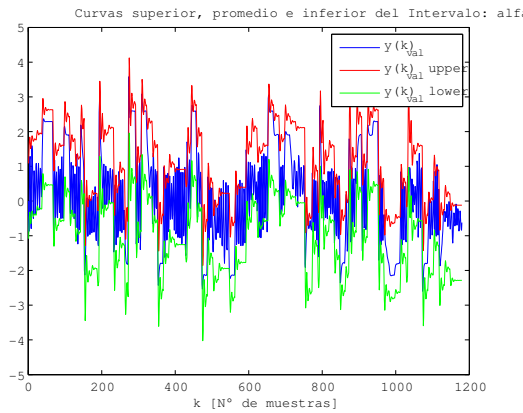


(a) Curvas para Intervalo del modelo lineal considerando predicción a 1 paso y $\alpha = 0,8$

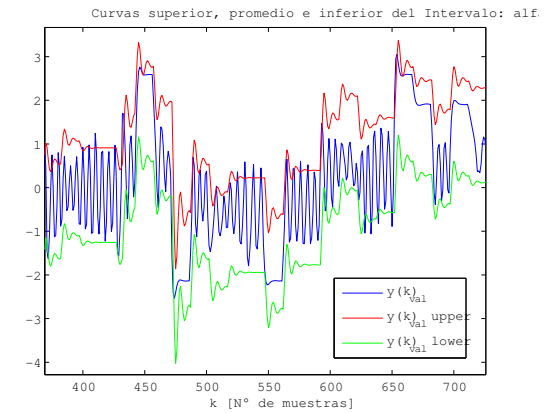


(b) Porción del gráfico

Figura 6: Curvas para Intervalo del modelo lineal considerando predicción a 1 paso y $\alpha = 0,8$



(a) Curvas para Intervalo del modelo lineal considerando predicción a 1 paso y $\alpha = 1,5$



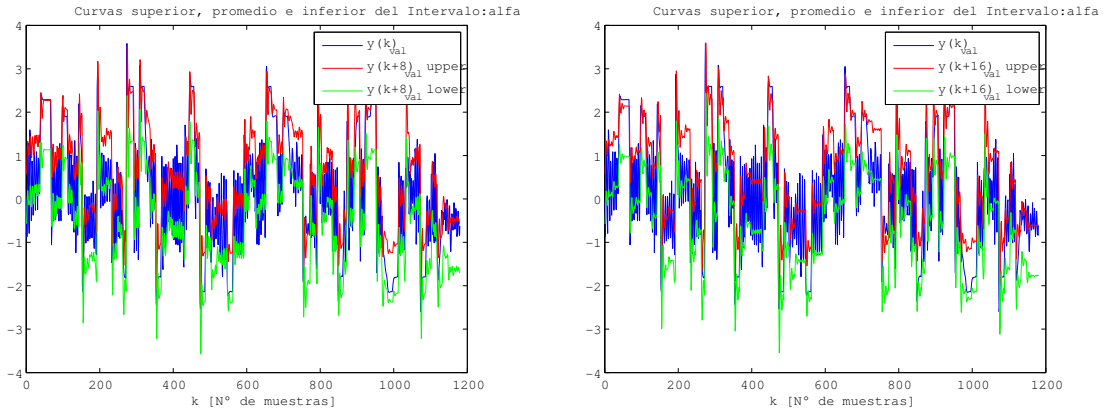
(b) Porción del gráfico

Figura 7: Curvas para Intervalo del modelo lineal considerando predicción a 1 paso y $\alpha = 1,5$

Ejercicio N°1

Es evidente, que los datos de salida de la curva promedio (color azul) están casi completamente contenidos en el intervalo (entre curva de color rojo y verde respectivamente), de hecho, está en un 88,56 % contenida en el intervalo. Ahora bien, en el caso de la construcción del intervalo de predicción a 8 y 16 pasos, se procede de manera análoga al caso de 1 paso, con la salvedad que se debe usar el modelo de 8 pasos y 16 pasos respectivamente.

A continuación, en la Fig. 8 se muestran los intervalos para el caso de 8 y 16 pasos considerando un parámetro $\alpha = 0,8$, para efectos de comparación con el gráfico de la Fig. 6.



(a) Curvas para Intervalo del modelo lineal considerando predicción a 8 pasos y $\alpha = 0,8$.

(b) Curvas para Intervalo del modelo lineal considerando predicción a 16 pasos y $\alpha = 0,8$.

Figura 8: Curvas para Intervalo del modelo lineal

Tabla 3: Métricas de porcentaje de cobertura y ancho del intervalo para diferentes valores de $\alpha = 0,8$ considerando intervalos de predicción a 1, 8 y 16 pasos respectivamente.

Parámetro $\alpha = 0,8$	0.8		
	1 pasos	8 pasos	16 pasos
PICP (%)	52.5424	52.7966	52.7966
PINAW	18.6992	18.4244	18.7348

Analizando las figuras anteriores y de la Tabla 3, se puede concluir que las métricas de bondad para los intervalos son bastante similares. Por una parte, entendemos que esto se debe a que como el número de muestras del conjunto es muy elevado (1.200 muestras aproximadamente), respecto a las predicciones que se piden (1, 8 y 16 pasos), es más o menos evidente que no exista diferencia en los porcentajes

de cobertura. La diferencia podría notarse en caso que los intervalos de predicción fueran a más muestras, por ejemplo, a 200 muestras en adelante. Otro factor que favorece la similitud de las métricas anteriores, es el hecho que la entrada (señal APRBS) es bastante aleatoria y por tanto el ajuste no se ve más favorecido en un conjunto que en otro (conjunto de prueba o validación, por ejemplo). Es decir, los parámetros de ajuste que se obtienen son similares, no importando que conjunto que usemos para entrenamiento. Es evidente que el caso del conjunto de entrenamiento, lo favorece el hecho de que cuenta con más datos, pero dada la aleatoriedad de la señal de entrada, este factor no es tan incidente a la hora de comparar las métricas.

3.2. Modelo difuso Takagi-Sugeno Tipo-1

Los modelos difusos de Takagi-Sugeno son estructuras basadas en la lógica difusa que permiten representar procesos con dinámicas no lineales mediante la combinación de información otorgada por modelos locales. Estos tipos de modelos pueden ser expresados a partir de una base de reglas del tipo “Si-Entonces”, de la forma

$$R_r : \text{Si } z_1(k) \text{ es } MF_1^r \text{ y...y } z_p(k) \text{ es } MF_p^r \text{ entonces } y_r(z(k)) = f_r(z(k)) \quad (11)$$

donde R_r denota la r -ésima regla del modelo difuso, con $r \in 1, \dots, N_r$ y N_r el número total de reglas; $y_r(z(k))$ es su consecuencia o modelo local; $z(k) = [z_1(k), \dots, z_p(k)]$ es el vector de premisas en el tiempo k , las cuales por lo general son regresores de la entrada y/o salida del sistema; $f_r(z(k))$ es una función de las premisas del modelo; y MF_i^r es el conjunto difuso (función de pertenencia) de la i -ésima premisa correspondientes a la r -ésima regla.

Sea $\mu_r(z_i(k))$ el grado de pertenencia de la i -ésima premisa $z_i(k)$ al conjunto difuso MF_i^r , donde $\mu_r(z_i(k)) \in [0, 1]$, siendo 0 cuando la premisa no pertenece en ningún grado al conjunto MF_i^r , y siendo 1 si pertenece completamente a dicho conjunto. Luego, se define el grado de activación de la r -ésima regla, $w_r(z(k))$, como

$$w_r(z(k)) = \text{oper}(\mu_r(z_1(k)), \dots, \mu_r(z_p(k))) \quad (12)$$

donde $\text{oper}(\cdot)$ puede ser el operador mínimo o el producto. Se denota $h_r(z(k))$ al grado de activación normalizado de la r -ésima regla, es decir,

$$h_r(z(k)) = \frac{w_r(z(k))}{\sum_{l=1}^{N_r} w_l(z(k))} \quad (13)$$

Ya definido el grado de activación de cada regla, la salida del modelo difuso, $y_{fuzzy}(k)$, está dada por una suma ponderada de cada modelo local por su grado de activación normalizado, de la forma

$$y_{fuzzy}(k) = \sum_{r=1}^{N_r} h_r(z(k)) * y_r(z(k)) \quad (14)$$

Los modelos difusos TS son una clase de sistemas no lineales, cuya formulación requiere definir una serie de variables que no se conocen a priori, por lo que una estructura adecuada para representar un sistema es desconocida y, como consecuencia,

un proceso de identificación debe ser llevado a cabo para determinar la estructura y cada uno de los parámetros del modelo [5].

- **Selección de variables:** Para seleccionar las variables que actúan como entrada al sistema difuso, se realiza un análisis de sensibilidad. Suponiendo una estructura del modelo inicial difuso con 8 variables de entrada $y(k-1), \dots, y(k-4), u(k-1), \dots, u(k-4)$. En la Fig. 9 se muestran los índices de las sensibilidades del modelo inicial para las 8 variables de entrada, comprobándose que las variables $y(k-4)$, $u(k-3)$ y $u(k-4)$ presentan menores índices de las sensibilidades, por lo cual no son incluidos en el modelo difuso. A pesar de que los regresores $y(k-3)$ y $u(k-2)$ presentan una sensibilidad similar se decidió escoger $u(k-1)$ ya que el mismo es un parámetro de la serie no lineal dinámica.

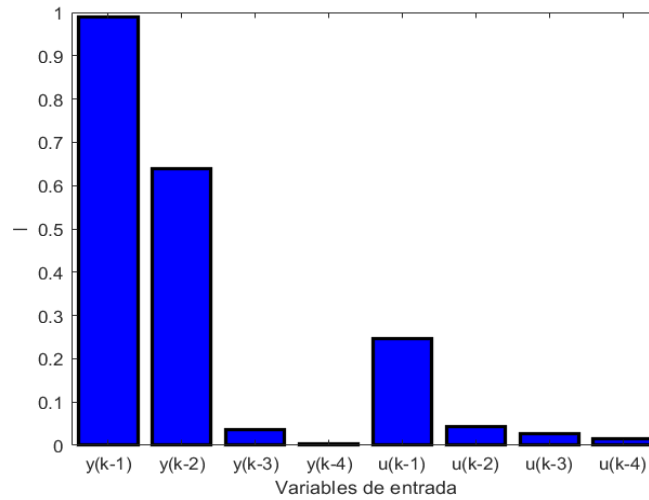


Figura 9: Índice de Sensibilidades.

La Tabla 4 indica el valor de la Raíz del Error Cuadrático Medio (RMSE) para los modelos con 8 y 4 regresores, y se puede notar que son muy semejantes, y por consiguiente se selecciona el modelo más sencillo, que coincide con el número de regresores de la serie no lineal original. Recordar que RMSE se define,

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^N (y(k) - y_{fuzzy}(k))^2} \quad (15)$$

donde N es la cantidad total de datos, $y(k)$ es la salida de la planta real en el instante k , y $y_{fuzzy}(k)$ es la predicción realizada por el modelo difuso en el instante k .

Tabla 4: Índices de Error para el Análisis de Sensibilidades.

Modelo	Variables de entrada	RMSE
1	$y(k-1), y(k-2), y(k-3), y(k-4)$ $u(k-1), u(k-2), u(k-3), u(k-4)$	0.2177
2	$y(k-1), y(k-2), u(k-1), u(k-2)$	0.2109

- **Optimización de la estructura:** La optimización de la estructura del modelo difuso consiste principalmente en determinar el número óptimo de reglas del modelo difuso. En este caso se definió un número máximo de 20 clusters y se entrenó el modelo para cada una de las posibles valores de clusters, utilizando como algoritmo de clustering el Fuzzy C-Means.

La Fig. 10 muestra el RMSE para los conjuntos de entrenamiento y prueba. Si no importa la complejidad, el mejor modelo es aquel que tiene menor RMSE. Sin embargo, es posible que un modelo con peor índice de desempeño, pero menos complejo que el modelo óptimo, pueda obtener resultados aceptables bajo un estándar de rendimiento definido preliminarmente. Por lo antes expuesto para este problema se escoge como número de clusters 5, por lo que el modelo difuso contará con 5 reglas, Fig 11.

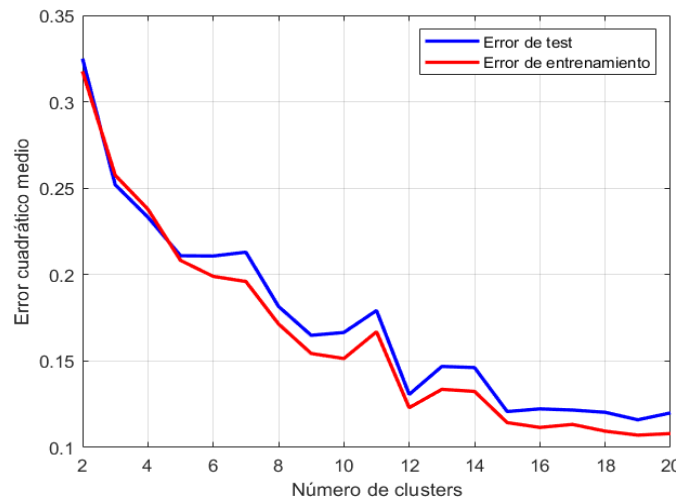


Figura 10: Índice de Sensibilidades.

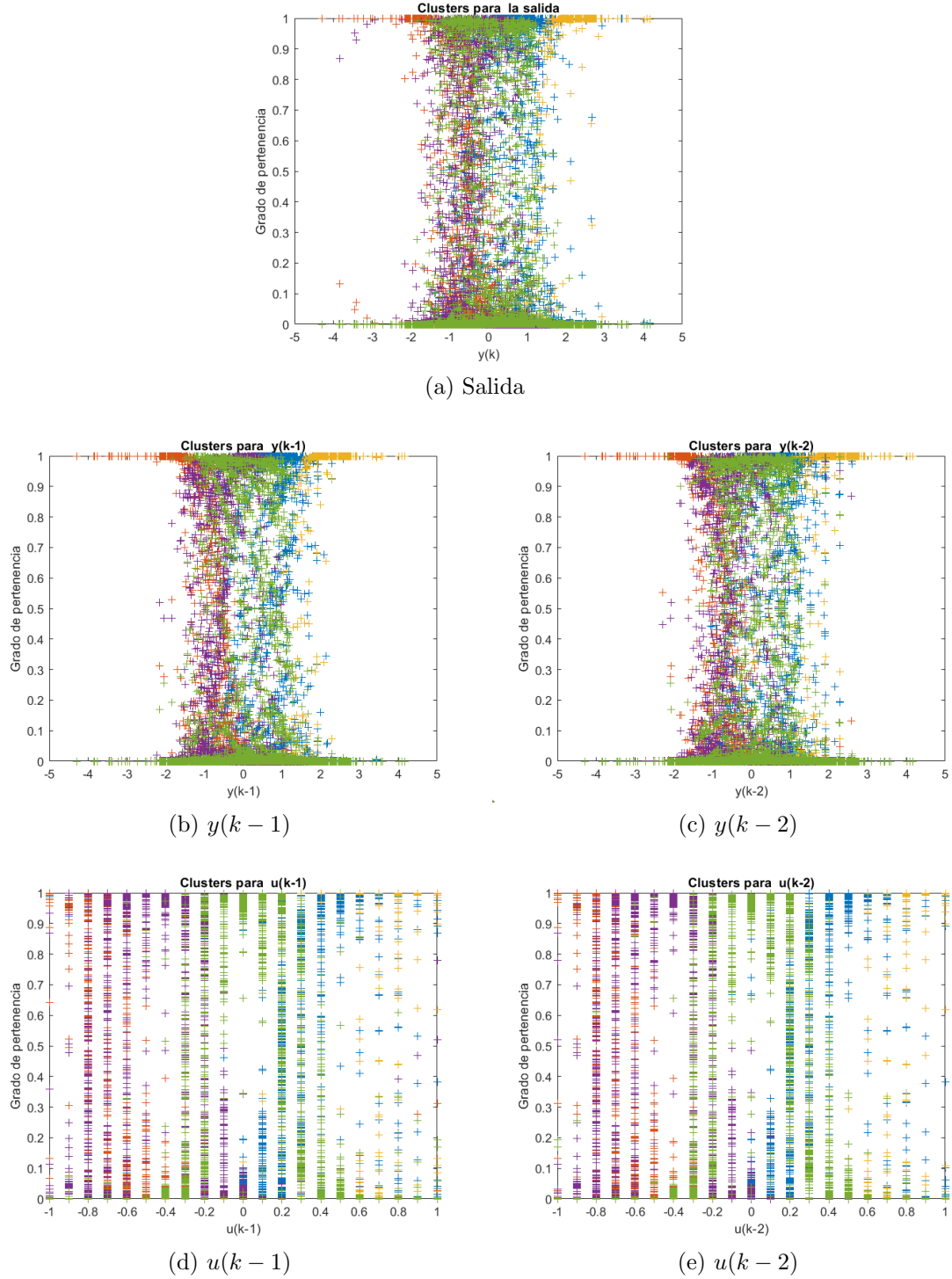


Figura 11: Clusters del Modelo Difuso Tipo 1.

Una vez realizado el clustering es posible proyectar las agrupaciones en el espacio de entrada y ajustar funciones paramétricas para describir los conjuntos difusos MF_i^r . En particular, se consideraron funciones de pertenencia gaussianas dadas por

$$MF_r^i(z_i(k)) = \exp(-0,5(a_{r,i} * (z_i(k) - b_{r,i}))^2) \quad (16)$$

donde $MF_r^i(z_i(k))$ es la función de pertenencia de la i -ésima premisa $z_i(k)$ al r -ésimo cluster, $a_{r,i}$ es el inverso de la desviación estándar de los datos ajustados por la gaussiana, $b_{r,i}$ representa su media. Para este modelo difuso se tiene:

$$a = \begin{pmatrix} 1,9799 & 1,4049 & 4,31509 & 4,9400 \\ 1,8300 & 1,4483 & 4,6810 & 6,1218 \\ 2,4547 & 1,8383 & 5,2820 & 8,5600 \\ 2,0375 & 1,773 & 4,5619 & 4,7213 \\ 1,6123 & 1,4681 & 4,9096 & 5,0399 \end{pmatrix} \quad (17)$$

$$b = \begin{pmatrix} 0,6742 & 0,4811 & 0,4425 & 0,4375 \\ -1,4657 & -1,4022 & -0,7580 & -0,7632 \\ 2,1665 & 2,1262 & 0,8501 & 0,86403 \\ -0,43602 & -0,5047 & -0,4782 & -0,4821 \\ -0,0353 & 0,2984 & 0,0744 & 0,07290 \end{pmatrix} \quad (18)$$

Además del RMSE, se pueden definir el Error Porcentual Absoluto Medio (MAPE, Mean Absolute Percentage Error) y Error Absoluto Medio (MAE, Mean Absolute Error) para comprobar el modelo difuso obtenido.

$$MAPE = \frac{1}{N} \sum_{k=1}^N \frac{y(k) - y_{fuzzy}(k)}{y(k)} \quad (19)$$

$$MAE = \frac{1}{N} \sum_{k=1}^N |y(k) - y_{fuzzy}(k)| \quad (20)$$

En la Tabla 5 se puede ver como el RMSE se mantiene de similar para los conjuntos de prueba y validación lo que indica la capacidad de generalización del modelo. Por su parte el índice MAPE, mide el tamaño del error (absoluto) en términos porcentuales, indicando que el error porcentual promedio del modelo se encuentra alrededor del 70 %, si bien este es un valor grande, es válido destacar que se está trabajando con una señal cuya salida se encuentra

Tabla 5: Índices de Error para el Modelo Difuso

Métricas	Conjuntos		
	Entrenamiento	Prueba	Validación
RMSE	0.2082	0.2109	0.1938
MAPE	70.92	65.65	67.71
MAE	0.1454	0.1505	0.1344

entre $[-2,3]$. El MAE tampoco presenta variaciones entre los diferentes conjuntos de datos. Considerando un compromiso entre complejidad y desempeño se determina que el modelo propuesto con 4 regresores y 5 reglas es suficiente.

- **Optimización de los parámetros:** Se utiliza el algortimo de clustering difuso Fuzzy C-Means obteniéndose siguientes parámetros de los consecuentes $\theta^T = [\theta_{1,1}, \dots, \theta_{N_r,1}, \dots, \theta_{1,n}, \dots, \theta_{N_r,n}]$

$$\theta^T = \begin{pmatrix} -0,1625 & 0,99755 & -0,7451 & 0,8430 & 0,2983 \\ 0,4561 & 1,1500 & -0,3126 & 0,5930 & 0,1362 \\ -0,2139 & 0,8892 & -0,2883 & 1,0148 & 0,2207 \\ -0,10934 & 0,8182 & -0,9810 & 0,6261 & 0,1752 \\ 0,0032 & 0,5939 & -0,9615 & 1,0893 & 0,3707 \end{pmatrix} \quad (21)$$

3.2.1. Predicciones a 1, 8, y 16 pasos

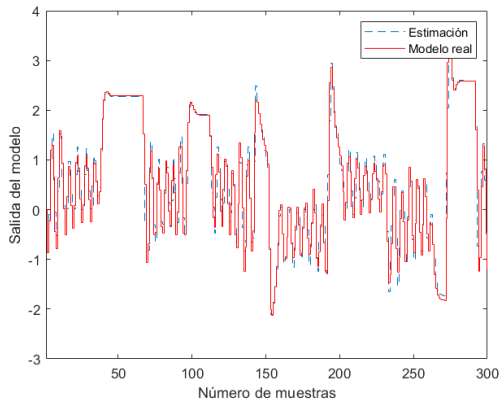
Las Fig. 12 muestra la salida del modelo difuso y la estimación para 1, 8 y 16 pasos respectivamente.

En la Tabla 6 se establece una comparación entre los modelos obtenidos, como se puede comprobar a medida que aumenta el paso de la estimación el desempeño del modelo difuso se deteriora, ya que el modelo comienza a realizar estimaciones a partir de sus propias estimaciones. Sin embargo entre la estimación a 8 y 16 pasos no existen marcadas diferencias, lo cual indica que se puede utilizar el modelo obtenido para predecir a j-pasos, siempre que se tenga en cuenta que se está trabajando con estimaciones.

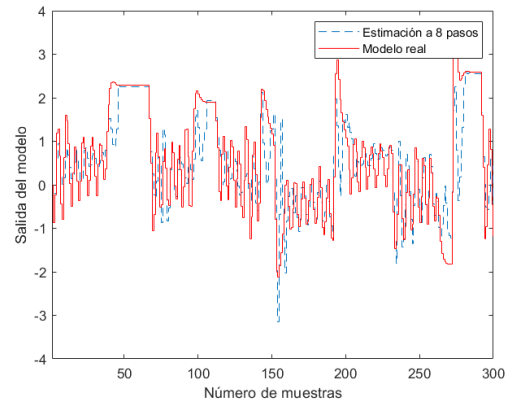
3.2.2. Intervalos de predicción

El modelo de intervalo difuso corresponde a un modelo difuso con parámetros superiores e inferiores que contiene un porcentaje de todos los valores medidos. Se debe encontrar una función difusa superior \bar{f} y una función difusa inferior \underline{f} , tal que se satisfaga

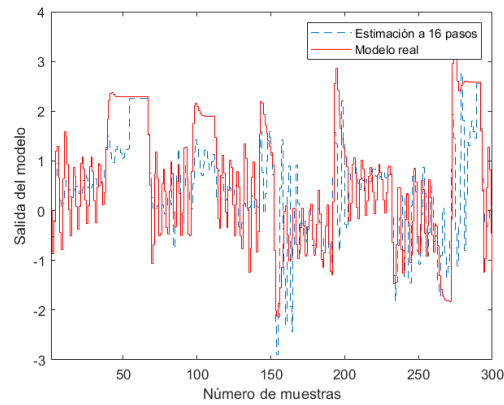
Ejercicio N°1



(a) Salida del Modelo a 1 paso



(b) Salida del modelo a 8 pasos



(c) Salida del modelo a 16 pasos

Figura 12: Respuesta del Modelo Difuso Tipo 1.

Tabla 6: Desempeño del Modelo difuso a j-pasos

Métricas	Modelos		
	Predicción a 1 paso	Predicción a 8 pasos	Predicción a 16 pasos
RMSE	0.1938	0.6740	0.8653
MAPE	67.71	147.36	186.32
MAE	0.1344	0.4747	0.6672

$$\underline{f}(z_k) \leq g(z_k) \leq \bar{f}(z_k) \quad (22)$$

en donde $z_k \in Z$ es un conjunto de entradas, $Y = y_1, \dots, y_N$ contiene los valores medidos de la salida, y se tiene que $y_k = g(z_k)$, $k = 1, \dots, N$.

Para encontrar las funciones \underline{f} y \bar{f} se utilizarán dos métodos: el método de la covarianza, y el método min-max.

El principal requerimiento al definir la banda del intervalo es que sea lo mas estrecha posible y que contenga un cierto porcentaje de datos, llamado nivel de confianza para lo cual se definen los índices Prediction Interval Coverage Probability (PICP) para medir el porcentaje de cobertura y el Prediction Interval Normalized Average Width (PINAW) para medir el ancho promedio.

$$PICP = \frac{1}{N} \sum_{k=1}^N c \quad c = \begin{cases} 1, & \hat{y}_l(k) \leq y(k) \leq \hat{y}_u(k) \\ 0, & \text{caso contrario.} \end{cases} \quad (23)$$

$$PINAW = \frac{1}{NR} \sum_{k=1}^N (\hat{y}_u(k) - \hat{y}_l(k)) \quad (24)$$

donde $R = \max(y(k)) - \min(y(k))$

■ Método de la Covarianza:

Este método se basa en utilizar la covarianza del error entre los datos reales y la estimación de los modelos locales del sistema difuso, de tal manera de determinar los parámetros de las funciones difusas limitantes a partir de cada consecuencia [6, 7].

$$Var(\hat{y}_j - h_j(Z^*)y_j) = \hat{\sigma}_j^2(1 + \Psi_j^{*T}(\Psi_j\Psi_j^T)^{-1}\Psi_j^*) \quad (25)$$

con la que se puede definir un intervalo difuso para cada modelo local

$$\hat{y}^j(k) = Z^{*T}(k)\hat{\theta}_j + \alpha[Var(\hat{y}_j - h_j(Z^*)y_j)]^2 \quad (26)$$

$$\hat{y}^j(k) = Z^{*T}(k)\hat{\theta}_j - \alpha[Var(\hat{y}_j - h_j(Z^*)y_j)]^2 \quad (27)$$

y obtener el modelo de intervalo total

$$\hat{y}_u(k) = \sum_{j=1}^{N_r} h_j(Z^*(k)) * \hat{y}_u^j(k) \quad (28)$$

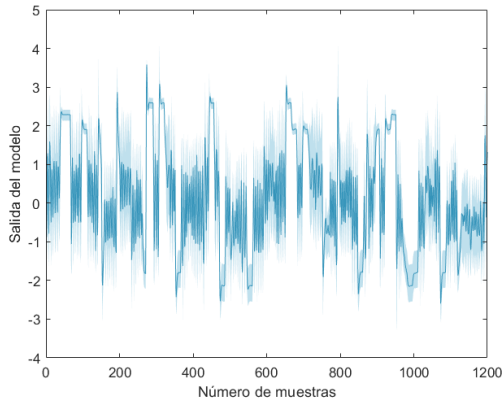
$$\hat{y}_l(k) = \sum_{j=1}^{N_r} h_j(Z^*(k)) * \hat{y}_l^j(k) \quad (29)$$

En la Tabla 7 se muestra un análisis para tres valores de α diferentes, como se puede observar $\alpha = 1$ da como resultado un ancho promedio porcentual del intervalo muy pequeño lo que implica un porcentaje de cobertura insuficiente para el modelo difuso, por lo cual esta opción no es válida. A medida que aumenta el valor de α aumenta el PICP pero también aumenta el PINAW, siendo necesario establecer un compromiso, en este problema es seleccionado el $\alpha = 10$ ya que presenta para la estimación a un paso 100 % de cobertura y para las estimaciones a 8 y 16 pasos valores superiores al 75 %, aunque esto implique un aumento del PINAW. En la Fig. 13 se muestra los intervalos difusos obtenidos.

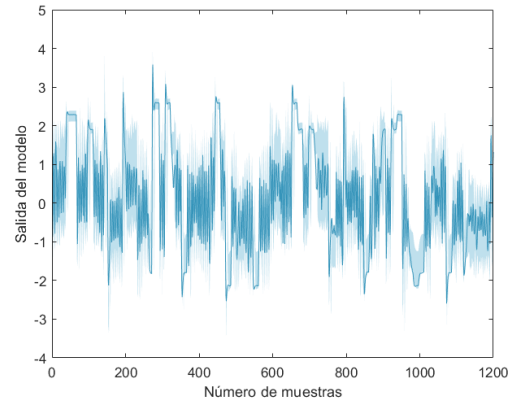
En las estimaciones a j-pasos los intervalos difusos cobran una mayor importancia, pues no solo se está prediciendo un valor, sino que además se indica con cual porcentaje de confianza dici valor se encuentra dentro de un intervalo de tal amplitud.

Tabla 7: Intervalos de predicción. Método de la Covarianza

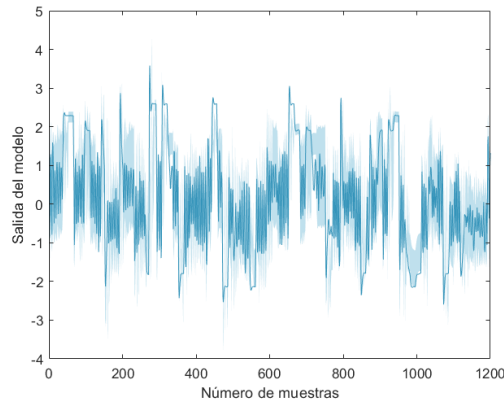
	$\alpha = 1$			$\alpha = 5$			$\alpha = 10$		
Índices	1 paso	8 pasos	16 pasos	1 paso	8 pasos	16 pasos	1 paso	8 pasos	16 pasos
PICP	39.5	14	9.08	97.5	63	47	100	86.5	76.25
PINAW	2.76	2.89	2.97	13.78	14.448	14.87	27.55	28.88	29.74



(a) Salida del modelo a 1 paso.



(b) Salida del modelo a 8 pasos



(c) Salida del modelo a 16 pasos

Figura 13: Intervalo difuso. Método de la Covarianza.

- **MinMax:** A diferencia del anterior no se fija un porcentaje de datos para el cual se desea el mínimo intervalo de confianza, sino que se realiza una optimización para que se contenga la mayor cantidad de datos dentro del intervalo de confianza y al mismo tiempo optimizando que su ancho sea el menor. Por lo que asumiendo la misma estructura de modelo obtenida anteriormente (número de reglas, número de regresores, y parámetros de los antecedentes), se calculan los parámetros de las consecuencias de ambos modelos (θ_u, θ_l) minimizando el máximo error absoluto de modelación mediante los siguientes problemas de optimización

$$\min_{\theta_u} \max |y(k) - \sum_{j=1}^{N_r} h_j(Z(k)) * (\theta_u^j)^T Z(k)| \quad (30)$$

$$s.a \quad y(k) - \sum_{j=1}^{N_r} h_j(Z(k)) * (\theta_u^j)^T Z(k) \leq 0$$

$$\min_{\theta_l} \max |y(k) - \sum_{j=1}^{N_r} h_j(Z(k)) * (\theta_l^j)^T Z(k)| \quad (31)$$

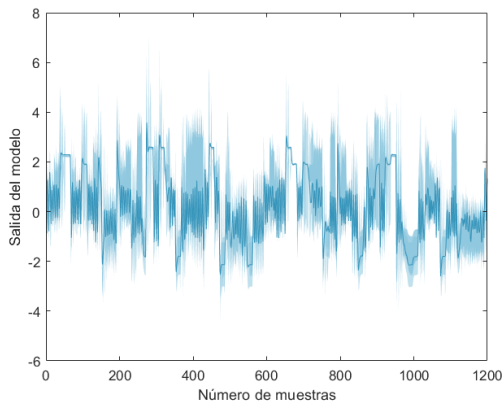
$$s.a \quad y(k) - \sum_{j=1}^{N_r} h_j(Z(k)) * (\theta_l^j)^T Z(k) \geq 0$$

En la Fig. 14 se muestra el resultado de la aplicación del Método MinMax, para los modelos difusos obtenidos, en este caso se tiene un porcentaje de cobertura inicial de 96.92%, al obtenerse a partir de un algoritmo de optimización no se puede predefinir este valor; para los modelos de predicción a j-pasos el porcentaje de cobertura disminuye. El ancho promedio porcentual del intervalo que se mantiene constante para los diferentes modelos obtenidos.

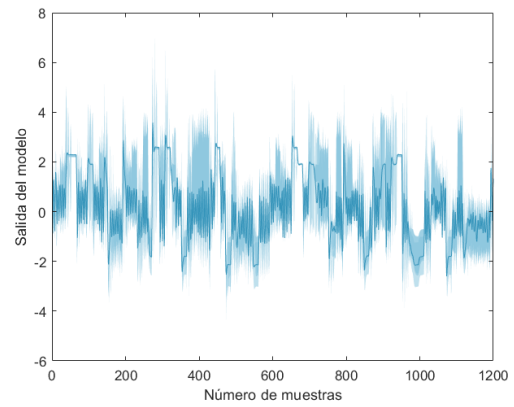
Tabla 8: Intervalos de predicción. Método MinMax

Métricas	Modelos		
	Predicción a 1 paso	Predicción a 8 paso	Predicción a 16 paso
PICP	92.58	77.25	69.92
PINAW	30.87	30.71	33.78

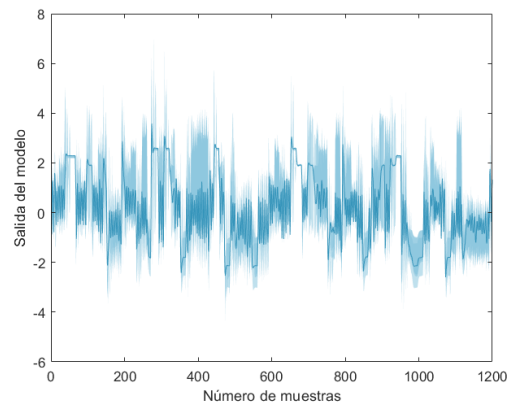
Si se comparan los métodos de intervalos difusos utilizados anteriormente, se puede determinar que el método de la Covarianza con un PINAW ligeramente inferior garantiza un mejor PICP para los modelos difusos de predicción a 1, 8 y 16 pasos.



(a) Salida del modelo a 1 paso.



(b) Salida del modelo a 8 pasos



(c) Salida del modelo a 16 pasos

Figura 14: Intervalo difuso. Método MinMax

3.3. Modelo de red neuronal

Para encontrar un buen modelo de red neuronal se propone seguir los pasos de identificación.

- **Obtención de datos:** Para ello se utiliza el set de datos creado en el punto a) del problema.
- **Selección de datos:** Análogo a los modelos anteriores, se utilizan los datos repartido con un 55 % en el conjunto de entrenamiento, 25 % en el conjunto de prueba y 20 % en el conjunto de validación.
- **Definición de la estructura de la red:** Se propone una red con una capa oculta, función de activación \tanh en la salida de la capa oculta y algoritmo de aprendizaje Levenberg-Marquardt. Las variables de entradas son iguales al número de regresores (4 entradas) como muestra la Figura 15. Por otro lado, el modelo matemático de la red, queda expresado como se muestra en la ecuación 32.

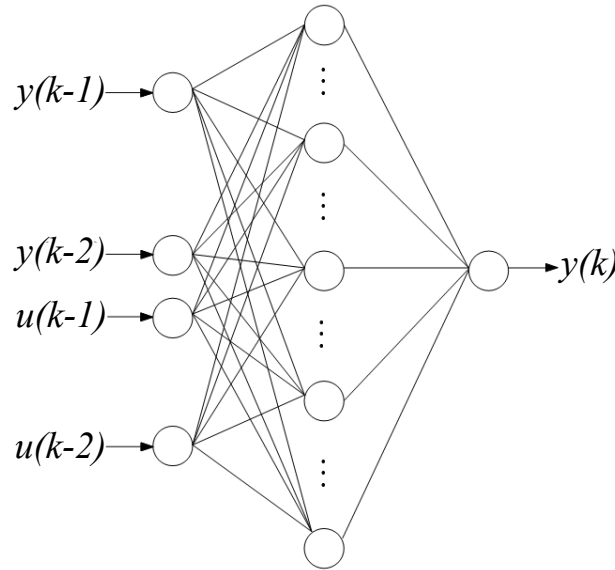


Figura 15: Estructura de la red neuronal perceptrón

$$\hat{y}(k) = \sum_{i=1}^{N_h} r w_i \left(\tanh \left(\sum_{j=1}^{N_I} l w_{ji} x_j + b_i \right) \right) + c \quad (32)$$

donde, $x = [y(k-1), y(k-2), u(k-1), u(k-2)]$ es el vector de entrada, N_h es el número de neuronas en la capa oculta, N_I es el número de variables en la

entrada, rw_i es el peso que conecta la i -ésima neurona de la capa oculta con el nodo de salida y lw_{ji} corresponde al peso que una la entrada j con la i -ésima neurona en la capa oculta. Los sesgos de cada neuronas en la capa oculta y para el nodo de salida son b_i y c respectivamente.

- **Selección de entradas relevantes:** De los 4 regresores presente en el sistema se debe analizar cuál tiene mayor peso en el modelo. Un método para encontrar dichos regresores es mediante un análisis de sensibilidad evaluando la derivada de la salida de la red por cada premisa de nuestros datos, es decir,

$$\xi_j = \frac{\partial \hat{y}(k)}{\partial x_j} \quad (33)$$

Como la funcion de activacion es \tanh y en la salida es lineal, se tiene,

$$\begin{aligned} \xi_j &= \frac{\partial \hat{y}(k)}{\partial x_j} \\ &= \sum_{i=1}^{N_h} rw_i \left(1 - \tanh \left(\sum_{m=1}^{N_I} lw_{mi} x_m + b_i \right)^2 \right) lw_{ji} \end{aligned} \quad (34)$$

Como se tendrá un valor de ξ_j para cada dato vector de entrada, se genera un vector $\boldsymbol{\xi}_j$ del mismo largo que el número de datos de cada variable.

Luego, se hace uso de un *indicador* I_j para cada entrada j definido como,

$$I_j = \mu^2(\boldsymbol{\xi}_j) + \sigma^2(\boldsymbol{\xi}_j) \quad (35)$$

donde μ es la media del vector de datos y σ^2 es la varianza para cada entrada j .

- **Optimización paramétrica y estructural:** Para encontrar los valores óptimos de las parámetros peso y sesgo de la red neuronal se utiliza el algoritmo de Levenberg-Marquardt backpropagation [8]. Por otro lado, para encontrar el óptimo de la estructura se analiza cuantas neuronas debe tener la capa oculta. Para ello se evalúa el RMSE (Raíz del error cuadrático medio) del conjunto de prueba en la salida de la red para un número de neuronas entre $[2 - 41]$. Los resultados de sensibilidad para cada neurona en la capa oculta se muestran en las Figura 17 y 18 y el RMSE evaluado en los 3 conjuntos se muestra en la Figura 16. Se puede ver que el mínimo RMSE para el conjunto de prueba es para 6 neuronas y que el modelo es menos sensible a la entrada $u(k-2)$ el cual es eliminado del entrenamiento.

El entrenamiento se configura a una velocidad inicial de aprendizaje de la red de 0,05 y un valor de épocas de 5000 con evaluación de *Overfitting* de 200 épocas de validación. Es importante señalar que durante el experimento se probaron diferentes épocas de validación para cuantificar el efecto del sobre ajuste en el número óptimo de neuronas con la hipótesis de que, independiente del RMSE obtenido, el mínimo global se da con la misma cantidad de neuronas óptimas afectando solo a los mínimos locales que pudieran aparecer. En la Tabla 9 se muestran los resultados de varios experimentos demostrando la hipótesis, es más, el valor RMSE obtenido con todo los modelos es casi el mismo y solo cambia el número de neuronas, donde, 6 demuestra ser un mínimo global y los demás un mínimo local. Por otro lado, se observo que tras varios experimentos el mejor valor RMSE del conjunto de prueba se observa cuando las épocas de validación por sobre ajuste se encuentran entre $[50 - 300]$, de esta manera, de aquí en adelante se utilizan 50 épocas de validación por sobre ajuste para disminuir los costos de calculo.

Tabla 9: Experimento para ver la influencia del sobre ajuste.

Número de épocas de validación	N_h óptimo	Valor RMSE
1	11	0.0025392
10	7	0.0025384
50	6	0.0025342
100	6	0.0025295
200	6	0.0025324
500	6	0.0025326
1000	10	0.0025299
5000	17	0.0025383

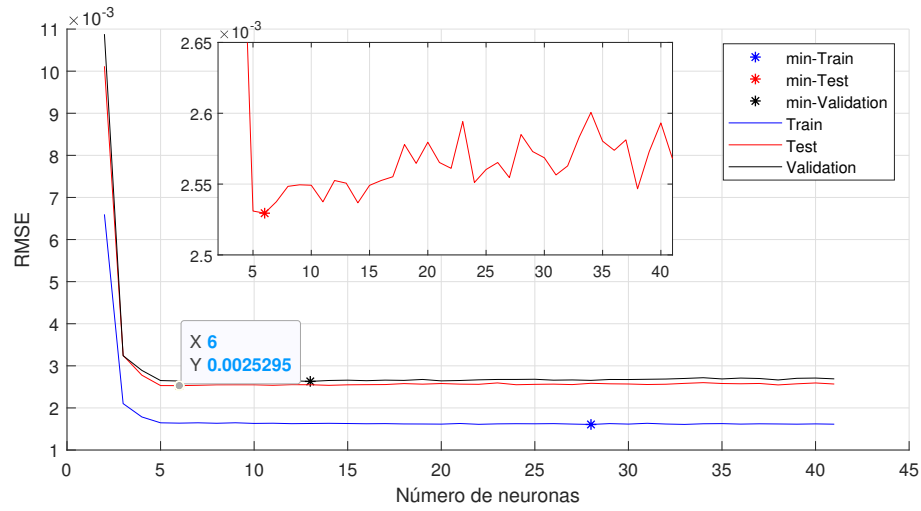


Figura 16: RMSE para diferente número de neuronas.

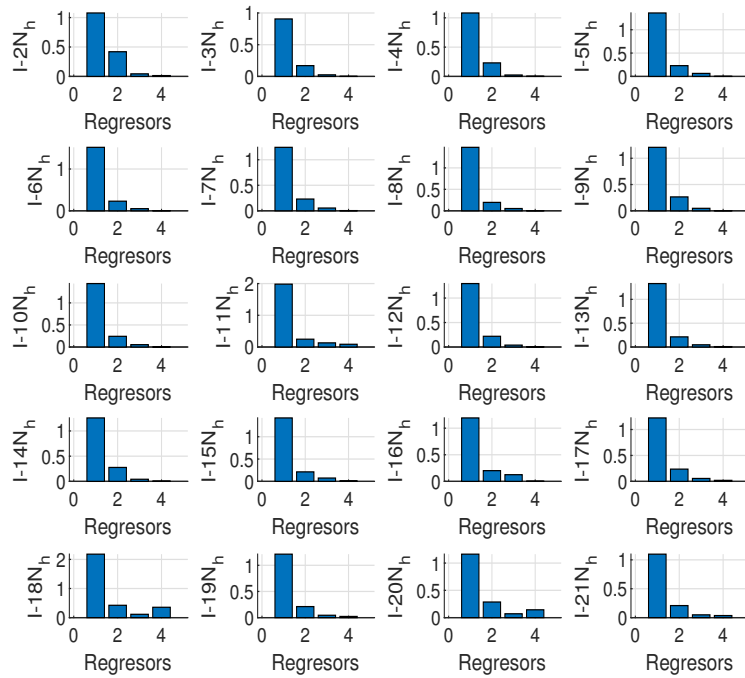


Figura 17: Sensibilidad para un número de neuronas entre $[2 - 21]$ en la capa oculta.

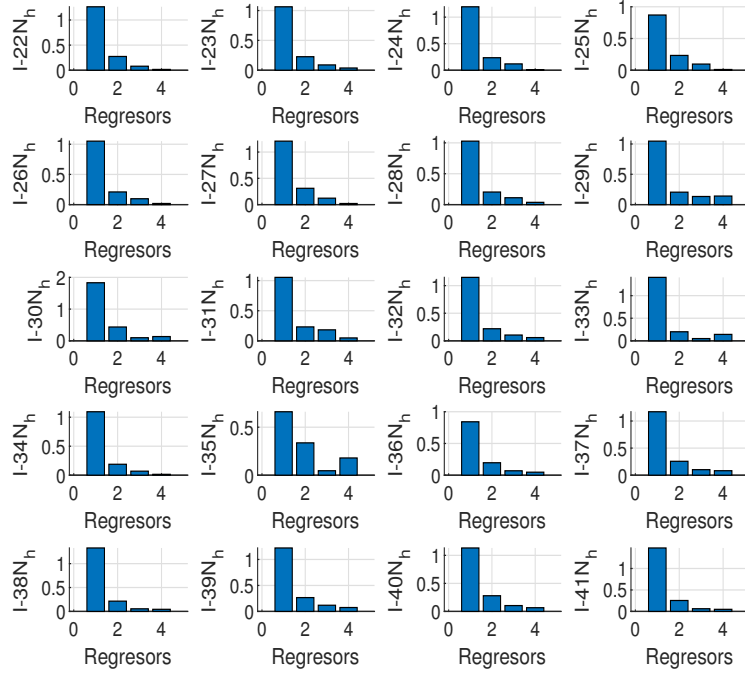


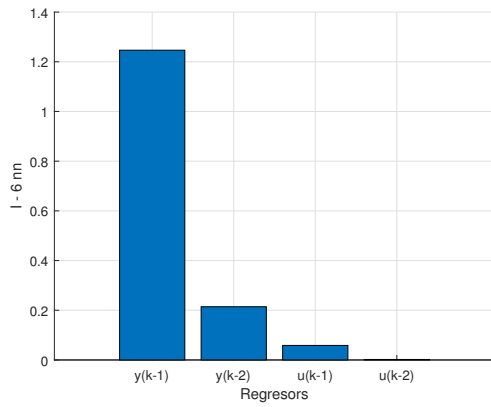
Figura 18: Sensibilidad para un número de neuronas entre $[22 - 41]$ en la capa oculta.

- **Desempeño de la red definida:** Se procede a evaluar el desempeño de la red con 8 neuronas en la capa oculta utilizando las cuatro entradas y luego solo con 3, $[y(k-1), y(k-2), u(k-1)]$, para notar las diferencias. Los resultados se muestran en las Figuras 19 hasta la 21 y los datos numéricos se agrupan en la Tabla 10

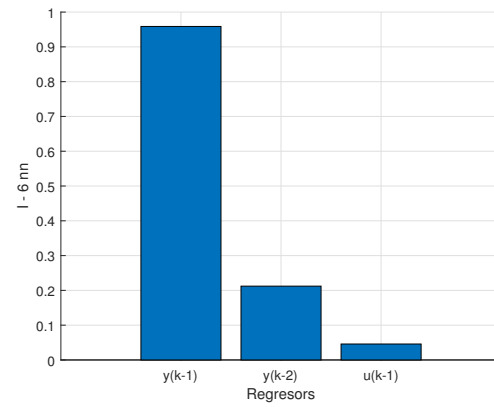
Tabla 10: Valores MSE para los 3 conjuntos de datos evaluados en una red con 4 entradas y otra con 3 entradas.

Número de entradas	MSE - Entrenamiento	MSE - Prueba	MSE - Validación
4	0.0089	0.0097	0.0084
3	0.0108	0.0118	0.0108

Se puede notar de la Figura 19 que al quitar la entrada $u(k-2)$ aumenta el indicador I_1 de la entrada $y(k-1)$ para compensar la falta del regresor $u(k-2)$ manteniendo casi al mismo valor las otras dos entradas. Por otro lado, no se eliminan más entradas dado que el ancho del histograma de la Figura 21(b) comienza a aumentar de valor mostrando que más datos tienen



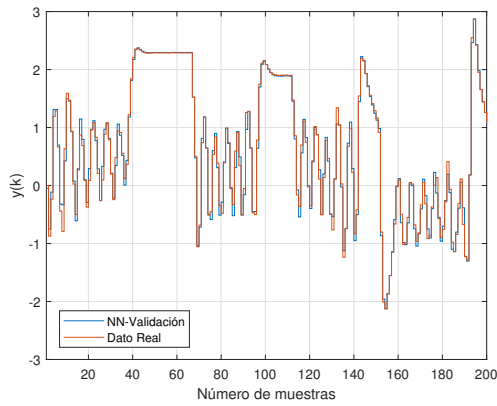
(a) 4 Entradas



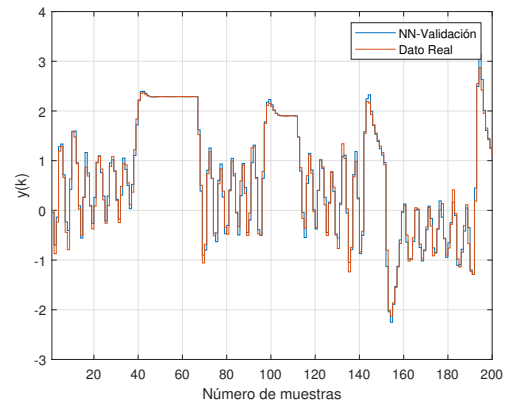
(b) 3 Entradas

Figura 19: Índice de sensibilidad para 4 y 3 entradas.

errores grandes. Este error puede tener solución al utilizar un nuevo número de neuronas en la capa oculta, por lo tanto, en el siguiente ítem se procede nuevamente a encontrar el número óptimo de neuronas con 3 entradas.

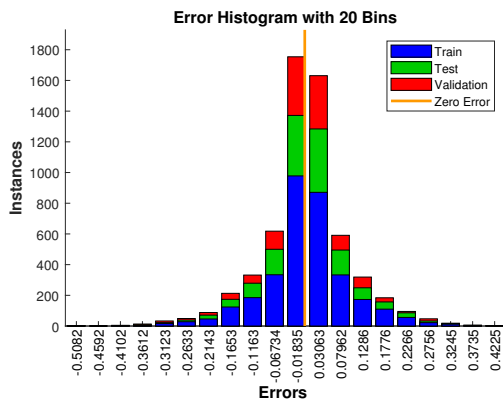


(a) 4 Entradas

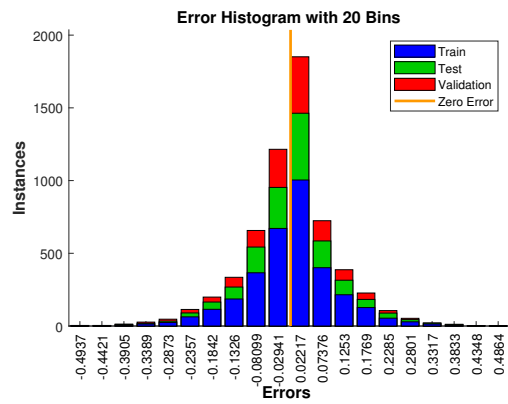


(b) 3 Entradas

Figura 20: Comparación entre la salida de la red neuronal y el valor real del conjunto de validación.



(a) 4 Entradas



(b) 3 Entradas

Figura 21: Histograma del error de cada conjunto de datos.

- **Optimizar estructura 2:** Se busca el número de neuronas óptimo en un rango de $[2 - 41]$. El resultado de la curva RMSE se muestra en la Figura 22 y el análisis de sensibilidad para cada neurona en las Figuras 23 y 24.

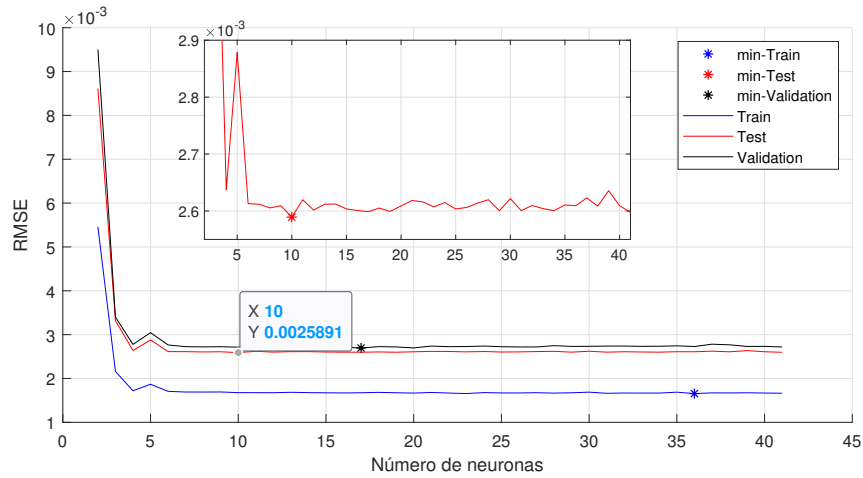


Figura 22: RMSE para diferente número de neuronas.

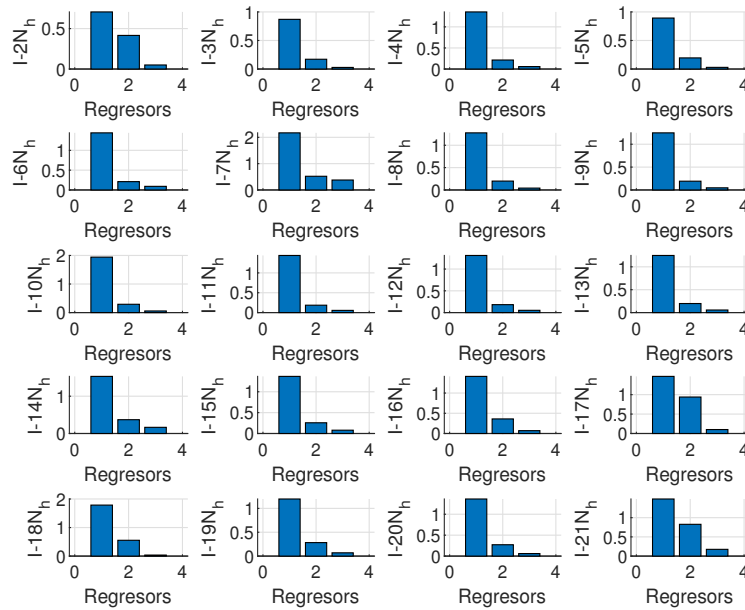


Figura 23: Sensibilidad para un número de neuronas entre $[2 - 21]$ en la capa oculta con 3 entradas.

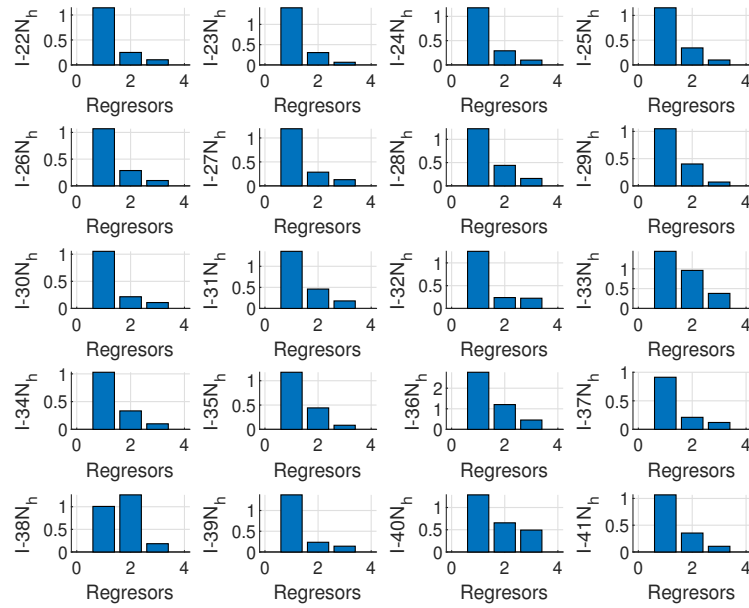
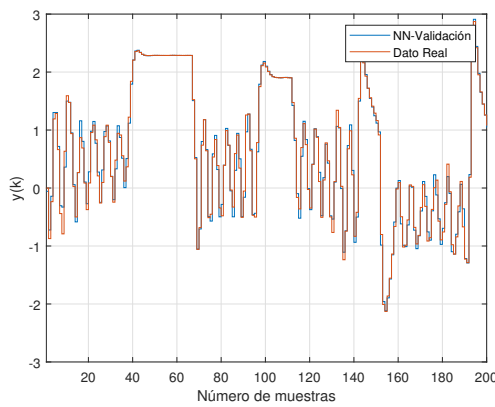


Figura 24: Sensibilidad para un número de neuronas entre $[22 - 41]$ en la capa oculta con 3 entradas.

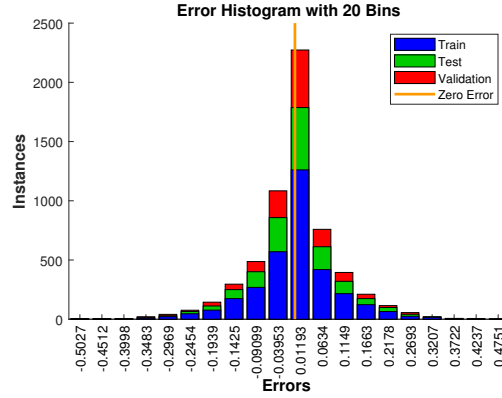
Validación final de la estructura: Para finalizar con la identificación del modelo de predicción por red neuronal, se evalúan los parámetros de la red, MAE, MAPE, RMSE, MSE reagrupados en la Tabla 11.

Tabla 11: Métricas finales del modelo neuronal

Métricas	Conjunto de entrenamiento	Conjunto de prueba	Conjunto de validación
RMSE	0.001681	0.002599	0.002726
MSE	0.009324	0.01013	0.008915
MAE	0.06538	0.06852	0.06117
MAPE	45.25	60.20	35.55

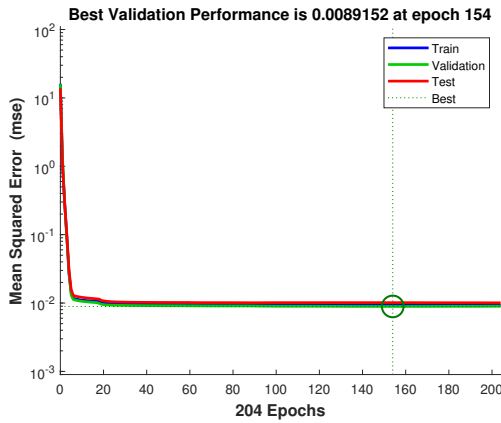


(a) RN vs datos reales

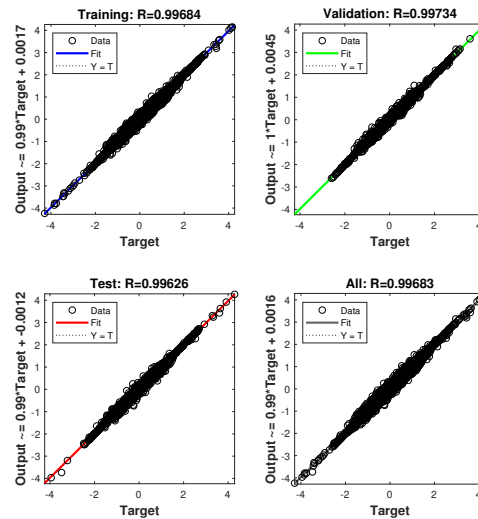


(b) Histograma del error

Figura 25: Resultados del modelo neuronal con 10 neuronas en la capa oculta en comparación con los datos reales.



(a) MSE de entrenamiento



(b) Regresión

Figura 26: Resultados de entrenamiento mostrando la regresión de cada conjunto.

3.3.1. Predicciones a 1, 8, y 16 pasos

Se comienza de la premisa que se conocen todos los valores hasta $\hat{y}(k)$.

- **A 1 paso:** Para realizar un modelo predictivo a 1 paso (un paso más y aparte

del generado en si por el modelo neuronal para pasar de $(k-1)$ a (k)), se tiene,

$$\hat{y}(k+1) = \sum_{i=1}^{N_h} r w_i \left(\tanh \left(\sum_{m=1}^{N_I} l w_{mi} x_m(k+1) + b_i \right) \right) + c \quad (36)$$

recordando la definición del vector x , se tiene, que $x(k+1) = [\hat{y}(k), y(k-1), u(k)]$. Existe una forma de determinar la señal de control futura $u(k)$ mediante modelo de control predictivo (MPC) [9], sin embargo, dado que se tiene un número grande de datos y que la señal APBRS se mantiene constante por lo menos en 40 puntos por bits antes de cambiar, se dará por conocido el control $u(k)$ requerido hasta el total de datos menos j . Los resultados para un paso se muestran en la Figura 27 para todos los conjuntos de datos.

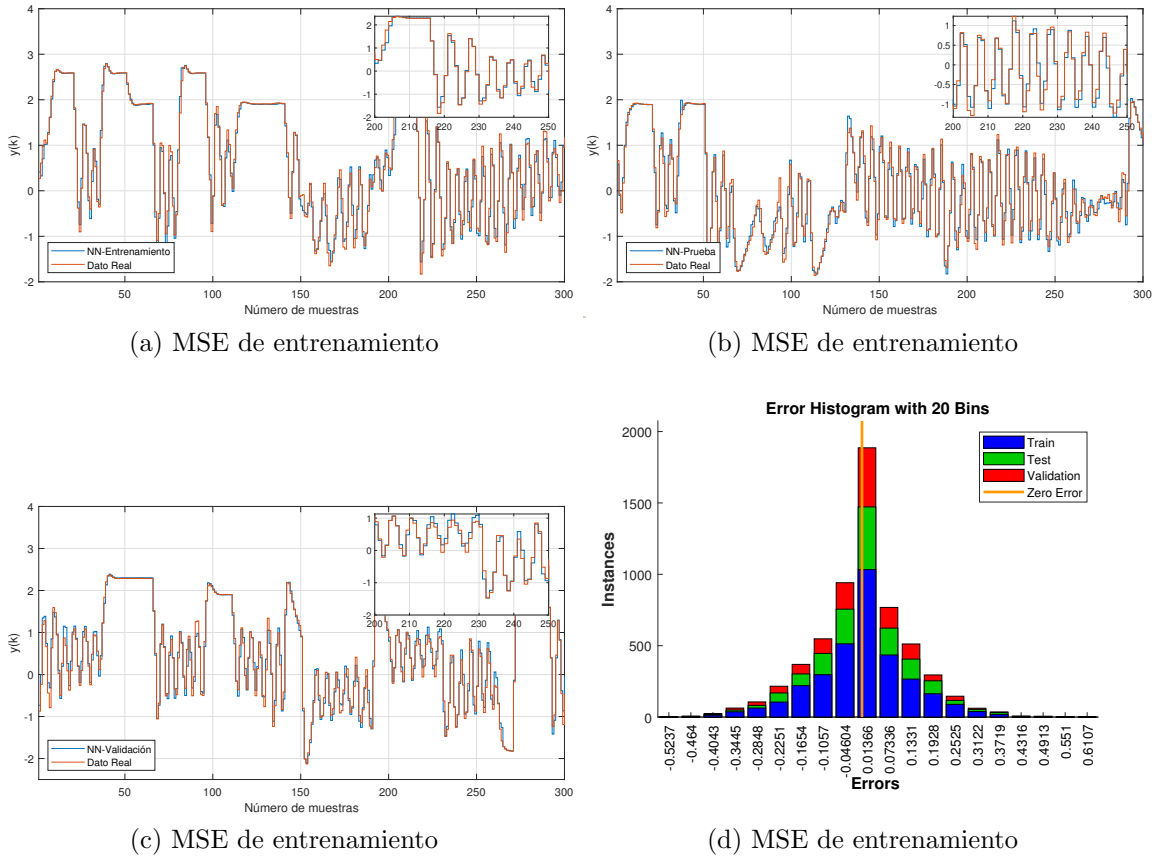


Figura 27: Predicción a 1 paso con modelo neuronal.

- **A 8 paso:** Para este escenario, junto al de 16 pasos, se requiere conocer entradas no conocidas. Estas entradas deben ser estimadas de los datos conocidos

como muestra la Figura 28, donde hasta $j = 1$ se conocen los datos y para $j \geq 2$ se deben usar las estimaciones anteriores. Los resultados para 8 pasos se muestran en la Figura 29.

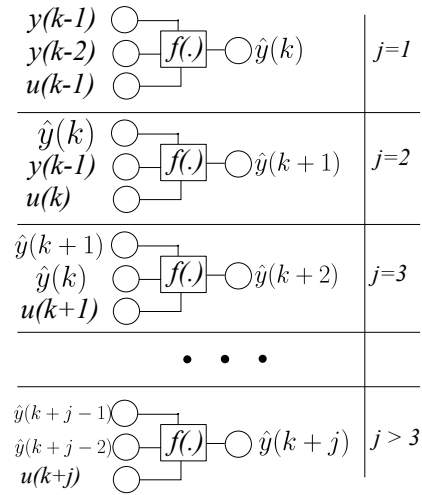


Figura 28: Modelo de predicción para j pasos del modelo neuronal.

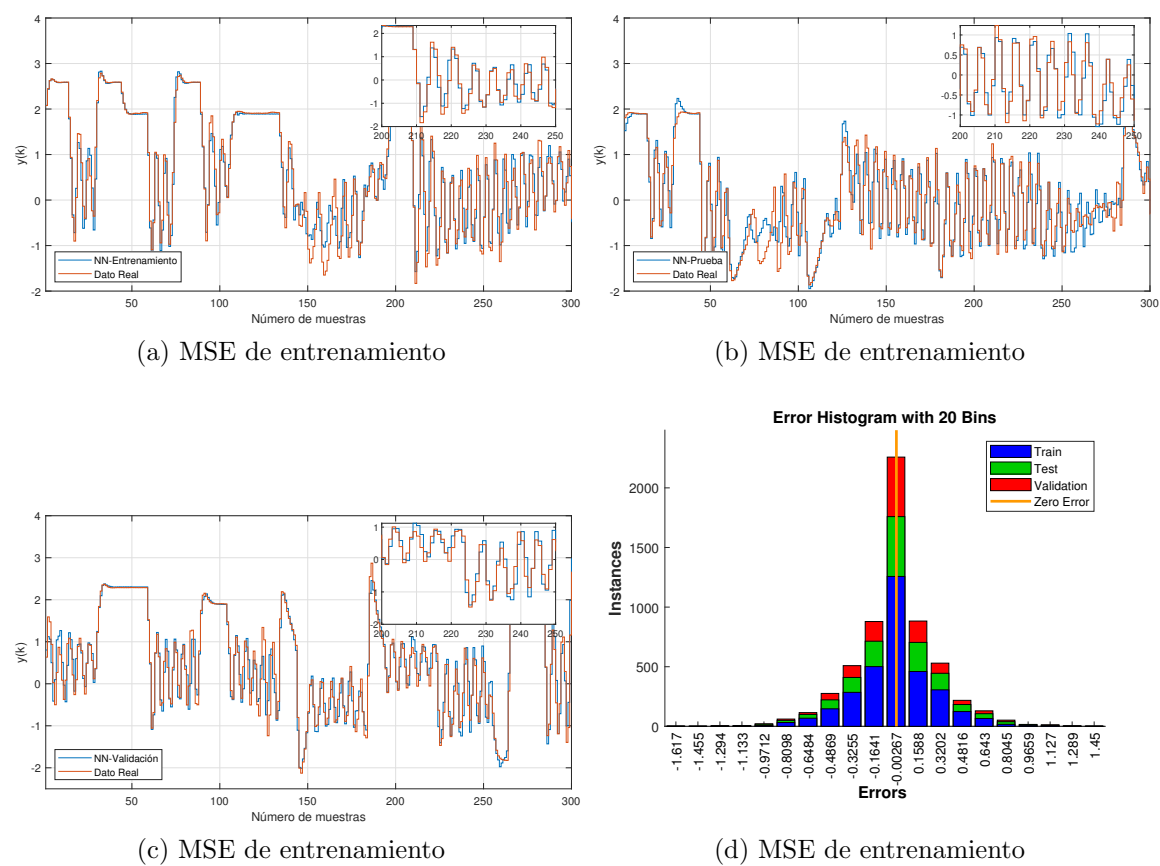


Figura 29: Predicción a 8 paso con modelo neuronal.

- **A 16 paso:** Los resultados se muestran en la Figura 30. Reagrupando las métricas, se tiene el resultado de la Tabla 12.

Tabla 12: Métricas de evaluación para 1, 8 y 16 pasos adelante.

Predicción a 1 paso			
Métricas	Conjunto de entrenamiento	Conjunto de prueba	Conjunto de validación
RMSE	0.0022	0.0034	0.0036
MSE	0.0162	0.0172	0.0157
MAE	0.0917	0.0948	0.0877
MAPE	61.1581	75.2533	37.2537
Predicción a 8 paso			
Métricas	Conjunto de entrenamiento	Conjunto de prueba	Conjunto de validación
RMSE	0.0050	0.0082	0.0082
MSE	0.0811	0.1012	0.0809
MAE	0.2006	0.2243	0.1900
MAPE	150.8271	172.2750	89.0900
Predicción a 16 paso			
Métricas	Conjunto de entrenamiento	Conjunto de prueba	Conjunto de validación
RMSE	0.0072	0.0122	0.0116
MSE	0.1695	0.2191	0.1582
MAE	0.2792	0.3272	0.2621
MAPE	209.8643	263.4764	125.4239

Comentarios: De las Figuras 27, 29 y 30 y la Tabla resumen 12, se puede ver que la red neuronal se ajusta muy bien a los datos reales manteniendo un error con distribución pseudo gaussiana (apartado (b) de las figuras) con un ancho bajo con error centrado en cero. Por otro, se nota que a medida que los pasos de predicción aumentan, las métricas comienzan a aumentar como es de esperarse, sobre todo el MAPE que mide la precisión de la predicción porcentual, es decir, puede que la curva de la red neuronal siga la curva gruesa de los datos reales sin necesidad de caer cerca de los datos reales, a pesar de esto, sigue siendo una predicción acertada. Se piensa que las perturbaciones y errores del modelo son pequeñas debido a que el tiempo de muestreo es muy alto y avanzar 16 pasos sea muy poco .

Ejercicio N°1

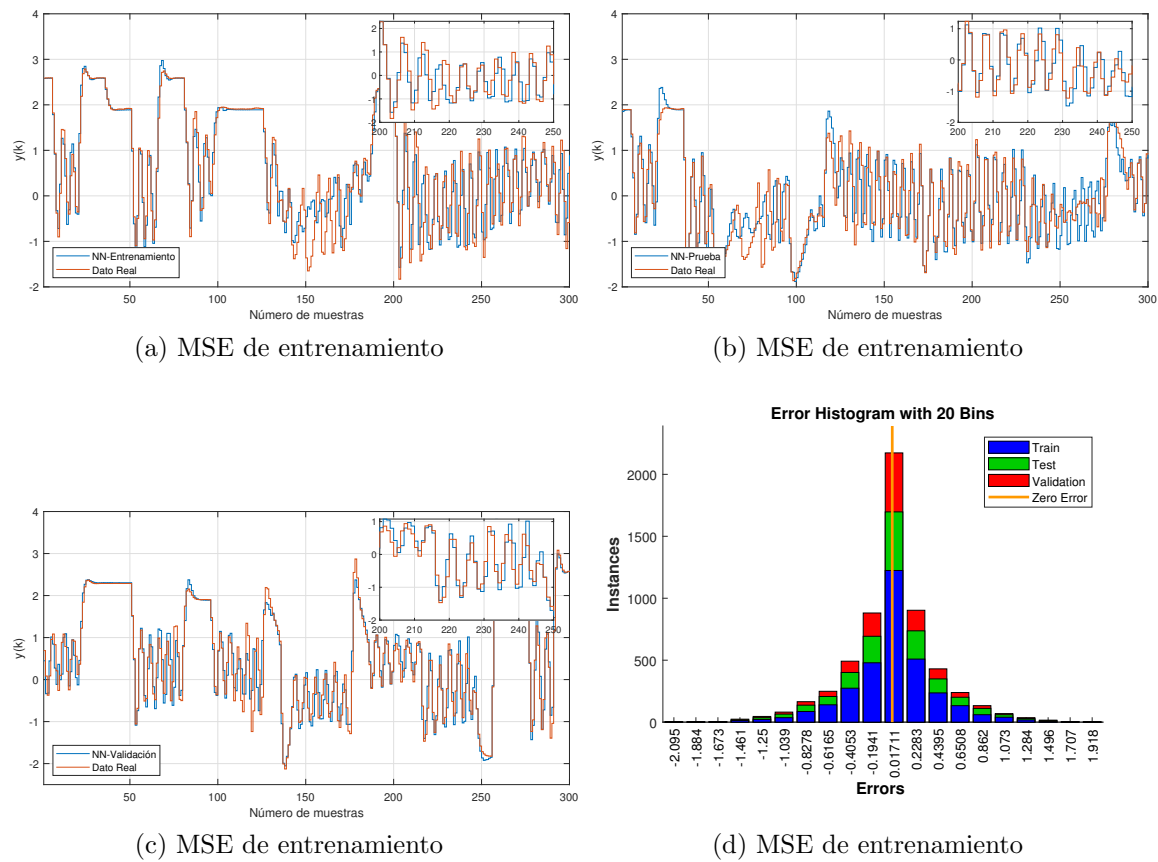


Figura 30: Predicción a 16 paso con modelo neuronal.

3.3.2. Intervalo de predicción

Para generar el intervalo de predicción en una red neuronal, se utiliza el método de la covarianza. Con este método se debe tener en cuenta algunos aspectos matemáticos que se desarrollan a continuación. Del modelo matemático de la ecuación 32, es posible reescribir la ecuación separando las capas, es decir,

$$\hat{y}(k) = \sum_{i=1}^{N_h} r w_i \tilde{Z}_i(k) + c \quad (37)$$

donde $\tilde{Z}_i(k)$ es la de la i -ésima neurona de la capa oculta, tal que,

$$\begin{aligned} \tilde{Z}_i(k) &= \tanh \left(\sum_{j=1}^{N_I} l w_{ji} x_j + b_i \right) \\ \tilde{Z} &= \tanh (LW X + B) \end{aligned} \quad (38)$$

Donde \tilde{Z} es la matriz de todos los datos a la salida de la capa oculta, X es la matriz que contiene todos los datos de la entrada, LW es la matriz de los pesos internos entre la entrada y la capa oculta y B es el vector de sesgo de cada neurona en la capa oculta. Luego, el método de la covarianza nos da los límites superiores $\hat{y}_u(k)$ y límite inferior $\hat{y}_l(k)$ como sigue,

$$\begin{aligned} \hat{y}_u(k) &= [RW \tilde{Z}^{*T}(k) + c] + \alpha \cdot \sigma_e^2 \left(1 + \tilde{Z}^{*T}(k) \left(\tilde{Z} \tilde{Z}^T \right)^{-1} \tilde{Z}^{*T}(k) \right)^{1/2} \\ \hat{y}_l(k) &= [RW \tilde{Z}^{*T}(k) + c] - \alpha \cdot \sigma_e^2 \left(1 + \tilde{Z}^{*T}(k) \left(\tilde{Z} \tilde{Z}^T \right)^{-1} \tilde{Z}^{*T}(k) \right)^{1/2} \end{aligned} \quad (39)$$

Donde,

- $\tilde{Z}^{*T}(k)$: Es el vector/matriz del conjunto de prueba
- \tilde{Z} : Vector/matriz de los datos de entrenamiento.
- RW : Matriz de pesos que conectan la capa oculta con la capa de salida de la red.
- α : Hiperparámetro para ajustar el ancho del intervalo.
- σ_e^2 : Varianza del error en el conjunto de entrenamiento $Var(y(k) - \hat{y}(k))$.

Notar que \tilde{Z} nace del conjunto de prueba, por lo tanto, para evaluar el intervalo de la varianza de validación, se deben calcular las mismas ecuaciones pero con el conjunto de validación. Por otro lado, para las predicción a j pasos, se debe calcular

Ejercicio N°1

la matriz de entrada de la red correspondiente a ese paso y calcular el valor de \tilde{Z} correcto. De esta forma, se obtienen los resultados mostrados en la Figura 31 los resultados del conjunto de validación. Las métricas PICP y PINAW se muestran en la Tabla 13

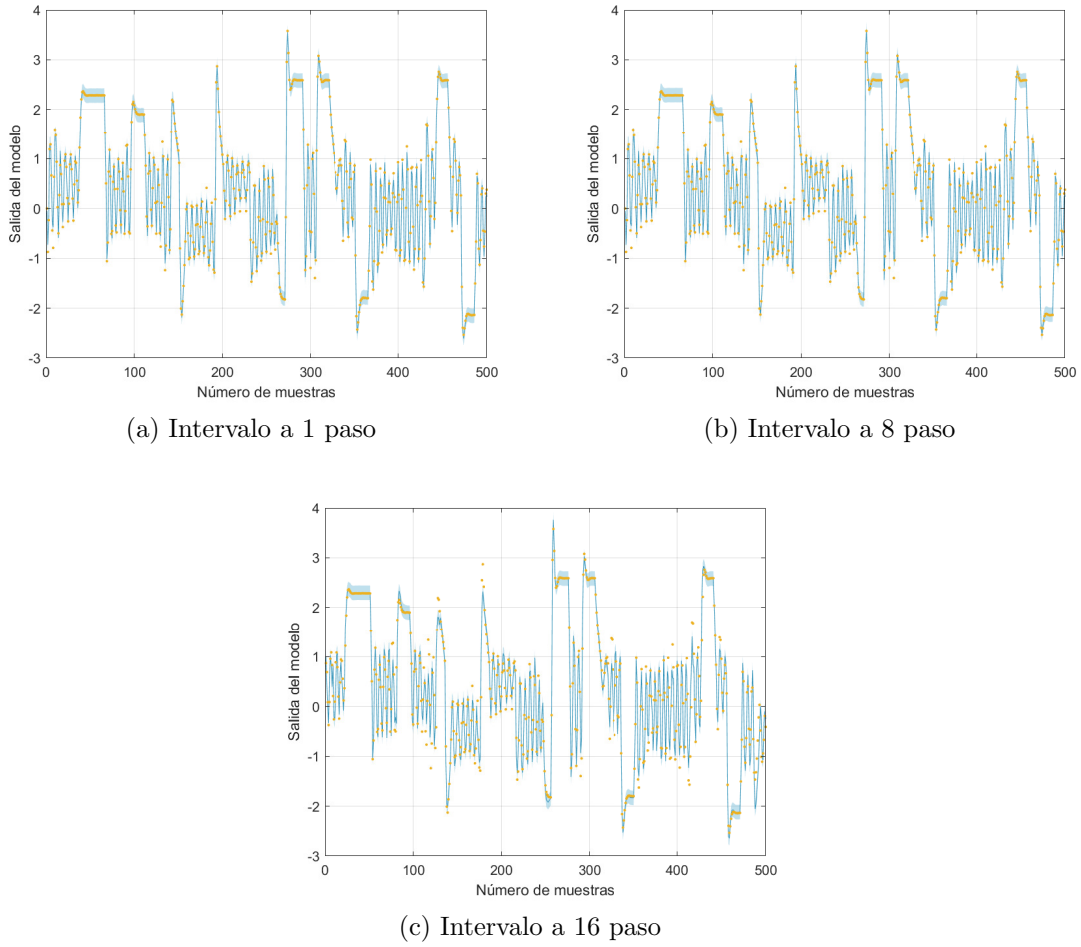


Figura 31: Uso del método de covarianza para estimar el intervalo de predicción a 1, 8 y 16 con $\alpha = 5$.

Tabla 13: Métricas de evaluación de intervalo para diferentes pasos con $\alpha = 5$.

Conjunto de validación			
	A 1 paso	A 8 pasos	A 16 pasos
PICP	88.0833	57.5859	48.1013
PINAW	4.6883	4.7021	4.7179

Ejercicio N°1

Como se puede apreciar en la Tabla 13, el PICP del primer paso no alcanza a cubrir todos los puntos reales del conjunto de validación, por ello, se presenta los resultados de otro escenario con $\alpha = 10$ en la Figura 32. Las métricas de evaluación de intervalo se reagrupan en la Tabla 14 donde se observa una mejora en el PICP con un leve aumento del PINAW.

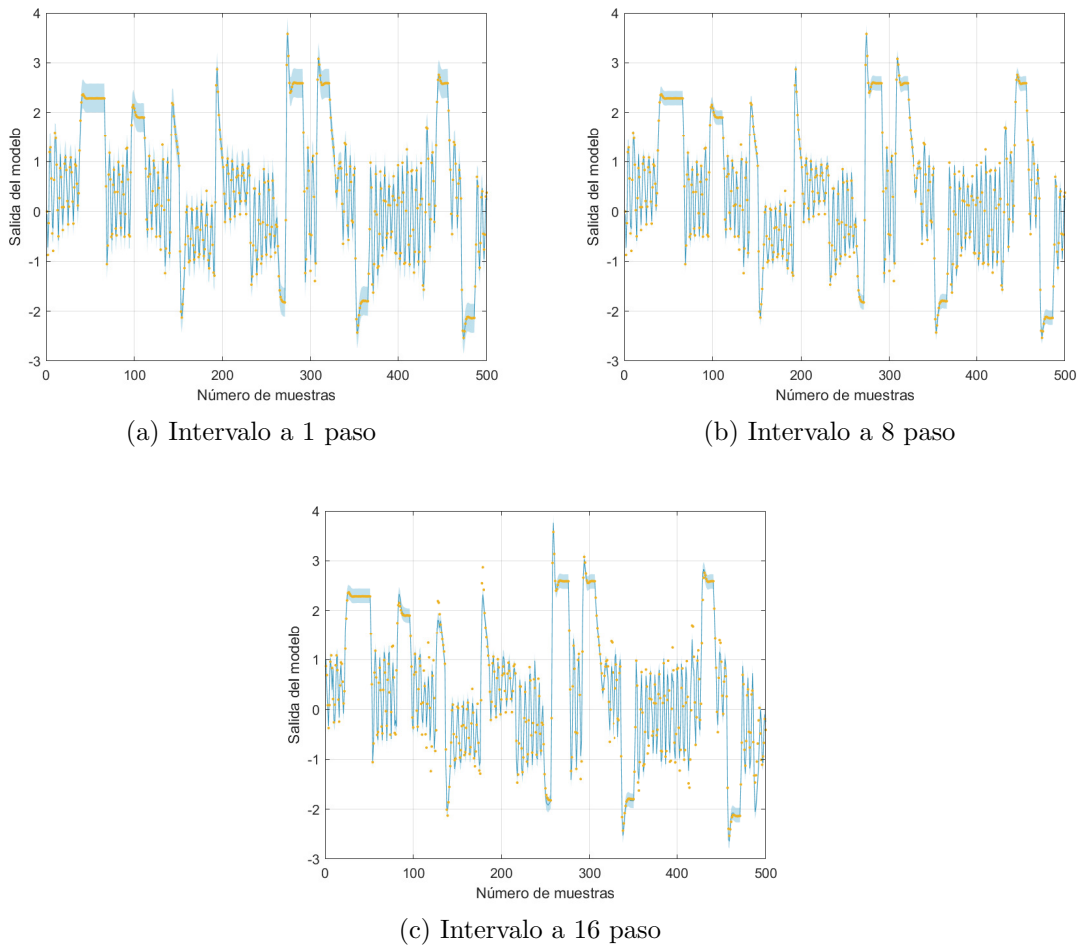


Figura 32: Uso del método de covarianza para estimar el intervalo de predicción a 1, 8 y 16 con $\alpha = 10$.

Tabla 14: Métricas de evaluación de intervalo para diferentes pasos con $\alpha = 10$.

	A 1 paso	A 8 pasos	A 16 pasos
PICP	98.3333	79.2121	68.4388
PINAW	9.3770	9.4056	9.4368

4. Resultados previos y conclusión

En la Tabla 15, se muestran las diversas métricas de bondad de predicción para los diferentes modelos analizados en este informe (Lineal, Takagi y Sugeno y Neuronal) usando los datos de validación.

Tabla 15: Comparación de los modelos propuestos

Modelos	Pasos	Parámetro de evaluación		
		RMSE	MAPE	MAE
Linal	1	0.0954	238.22	0.6020
	8	0.0789	226.60	0.5805
	16	0.0964	235.33	0.5944
T&S	1	0.1938	67.71	0.1344
	8	0.6740	147.36	0.4747
	16	0.8653	186.32	0.6672
RN	1	0.0036	37.2537	0.0877
	8	0.0082	89.0900	0.1900
	16	0.0116	125.4239	0.2621

Un análisis para intervalos difusos se presenta con el Método de la Covarianza (común para los tres modelos utilizados) en la Tabla 16

Tabla 16: Comparación de modelos de intervalo.

Modelos	Pasos	Parámetro de evaluación	
		PICP	PINAW
Linal	1	52.542	18.6992
	8	52.797	18.4244
	16	52.797	18.7348
T&S	1	100	27.55
	8	86.5	28.88
	16	76.25	29.74
RN	1	98.3333	9.3770
	8	79.2121	9.4056
	16	68.4388	9.4368

Se puede apreciar que, como el modelo de la planta es no lineal, la red neuronal predomina en todos los sectores de evaluación. El RMSE es bajo incluso para 16 pasos y el valor más alto de PINAW es alrededor de 9.5, menor que los otros modelos.

5. Intervalo de predicción con Joint Supervision

En esta sección se presenta el desarrollo de intervalos de predicción con Joint Supervision [10]. Para facilitar la implementación se procede a programar en python una red neuronal como se muestra en [3] en el archivo JoinSupervision.py. La lógica del código sigue la estructura presentada por [10] mostrada en la Figura 33, donde se debe buscar el mejor valor de λ para aumentar el PICP. Las ecuaciones no se presentaran ni se desarrollaran en este trabajo ya que se pueden ver fácilmente en la referencia, pero si se presenta el método aplicada para encontrar λ , el cual consiste en,

- Se entrena la red para paso igual 1
- Con el resultado se estima la predicción a J pasos
- Se evalúa el PICP de la salida (k+j)
- Si no cumple la condición de PICP se aumenta el valor de lambda.
- Se vuelve al punto inicial y se itera hasta encontrar el PICP deseado.
- Cuando se obtiene el PICP deseado, se mantiene constante el valor de λ y se corre el entrenamiento hasta maxCounter para encontrar el mínimo valor de PINAW.

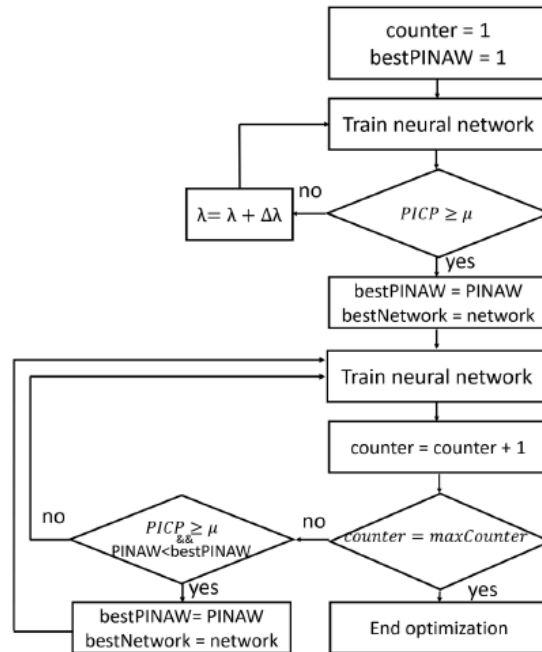
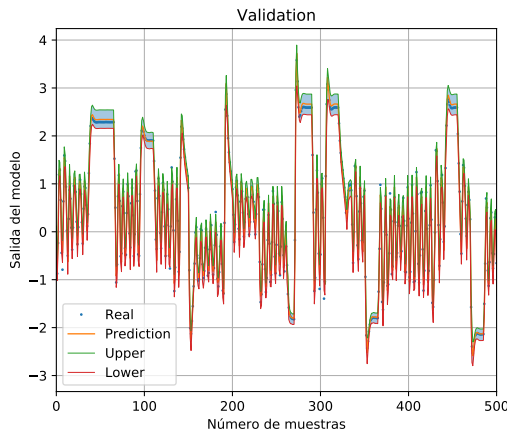


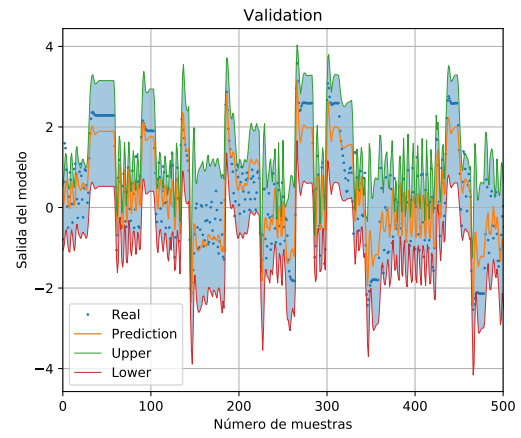
Figura 33: Metodología para desarrollar intervalos de predicción

Ejercicio N°1

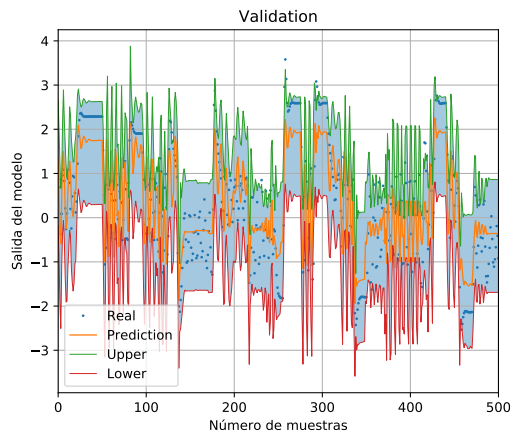
Para desarrollar el experimento y comparar los intervalos de predicción por covarianza, se genera un entrenamiento y optimización de λ con 4 regresores al usar Joint Supervision. Esto es porque en el entrenamiento de la nueva estructura de la red, la función de pérdida a sido modificada y esto puede afectar el modelo y cambiar los resultados del análisis de sensibilidad haciendo preferible el uso de todos los regresores. La optimización recae en 12 neuronas en la capa oculta con un $RMSE = 0,0968$ en el conjunto de prueba. Se propone un $\mu = 90\%$. Los resultados para 1, 8 y 16 pasos se muestran las Figuras 34 respectivamente, reagrupando las métricas en la Tabla 17.



(a) Intervalo a 1 paso



(b) Intervalo a 8 paso



(c) Intervalo a 16 paso

Figura 34: Uso del método de covarianza para estimar el intervalo de predicción a 1, 8 y 16 con $\alpha = 10$.

Tabla 17: Métricas de evaluación de intervalo para diferentes pasos con $\alpha = 10$.

	A 1 paso	A 8 pasos	A 16 pasos
PICP	90.399	92.3721	91.8143
PINAW	4.34012	41.3193	42.8199
MAE	0.077826	00.5676	0.60138
RMSE	0.1081	0.6739	0.7213

Se puede ver una mejora en el PICP con un bajo RMSE. Sin embargo, a diferencia de los resultados presentados en [10], los mostrados en la Tabla 17 muestran un PINAW más amplio. Las razones de por que puede pasar esto son variadas, entre ellas:

- La estructura de entrenamiento, es decir, definir bien si cada función de pérdida de cada salida debe entrenar por separado los pesos asociados a la conexión de la salida de la capa oculta y dicha salida (método de salidas independiente con solo las capas ocultas en común). Por otro lado, está el método que se basa en sumar las pérdidas de las 3 salidas y optimizar en base una función de pérdida común para las 3 salidas, lo que conlleva a que cualquier error en una de las salidas, genere un cambio en todos los pesos [11]. En el desarrollo de este problema se probaron ambos métodos mostrando mejores resultados el primero y es el que se utilizó finalmente.
- El uso de early stopping depende de que función de pérdida estamos evaluando (upper, crisp, lower). Se puede elegir una de ellas o la suma de las 3, cualquiera de las dos tiene un impacto en las otras 3 salidas.
- Existe la hipótesis de que los valores de la señal APBRS pueden influir en los resultados. Usar una señal con una amplitud máxima unitaria puede acotar la función de salida a valores pequeños con poca diferencia entre sí dificultando el entrenamiento de la red neuronal y sobre todo en las predicciones.
- Durante el ensayo de hiperparámetro se observó que el entrenamiento es muy ruidoso al cambiar la tasa de aprendizaje de 0.01 a 0.05. Para evitar esto se cambió a 0.005, sin embargo, para $j = 16$ pasos, vuelven a aparecer ruidos en el MSE del entrenamiento. Esto influye en el early stopping, pudiendo para antes solo por la existencia de ruido. Finalmente, el valor del incremento de λ puede afectar que tan grande nos da el siguiente PINAW, si se escoge un incremento muy pequeña, el código se demora mucho, si es muy grande, el PINAW puede quedar sub-ajustado y por eso puede que de tan grande para pasos más grandes.

De esta forma, el método es muy bueno para asegurar un PICP deseado, pero a

Ejercicio N°1

cambia, debe definirse muy bien toda su estructura y los puntos ya mencionados para evitar ambigüedades y que sea robusto a los valores aleatorios iniciales de los pesos, al ruido de entrenamiento, y lo que se define como función de pérdida total L_{total} .

6. Problema 2

Para la operación óptima de las micro-redes es importante contar con modelos de predicción confiables de variables tales como: potencia solar, potencia eólica, consumo, estado de carga de las baterías, entre otras variables. Los modelos e intervalos de predicción son importantes debido a la incertidumbre asociada a la generación con energía renovable y la variabilidad del consumo local.

Por tal razón, se le ha confiado a usted el proyecto de determinar modelos de predicción para la generación de energía en un sistema fotovoltaico instalado en una cierta comunidad del norte del país. La finalidad de este proyecto es la de disponer de información futura de generación fotovoltaica, para así poder gestionar el funcionamiento del resto de los elementos que componen a la micro-red.

Para esto se le entregarán datos históricos de generación fotovoltaica (expresada en KW) medida en la comunidad durante el periodo septiembre-diciembre de los años 2015 y 2017. Estos datos tienen un tiempo de muestreo de 1 hora (considerar porcentajes adecuados de los datos en las fases de training, test y validación).

Como usted sabe, son varias las formas que se pueden emplear para la modelación a partir de estos datos, por lo que debe seleccionar el tipo de modelo más adecuado para dicha aplicación.

Se sugiere considerar los datos del año 2015 como base de entrenamiento y las mediciones del año 2017 como base de prueba y validación. Para este trabajo se le pide detallar la metodología utilizada para:

- a) Obtener dos modelos de predicción (a elección entre los vistos en este curso) y evaluar las predicciones a 1, 6 y 12 pasos. Comparar el desempeño de todos los modelos a partir de las métricas más apropiadas tales como RMSE, MAPE, MAE, entre otras.
- b) Construir un intervalo de predicción para los modelos obtenidos en a), utilizando método que ud. seleccione.

6.1. Selección de Datos

Se cuenta con los datos históricos de generación fotovoltaica (expresada en KW) medida en la comunidad durante el periodo septiembre-diciembre de los años 2015 y 2017. Para la selección de los conjuntos de entrenamiento, prueba y validación se utiliza una división [60 %, 20 %, 20 %], tomándose en este caso 2880 muestras del año 2015 para entrenamiento, que corresponde a 120 días de muestreo, y para prueba y validación 960 muestras del año 2017, el equivalente a 40 días.

6.2. Modelo de red neuronal

Se comienza con un modelo de la red inicial usando 24 regresores para buscar el número óptimo de neuronas.

Referencias

- [1] D. S. H., “Apuntes el7012 – control inteligente de sistemas, modelación difusa,” 2016.
- [2] —, “Apuntes el7012 – control inteligente de sistemas, modelación neuronal,” 2016.
- [3] E. Obrequé, “El7012-tarea1.” [Online]. Available: <https://github.com/EliasObrequé/EL7012-Tarea1>
- [4] O. Nelles, *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*. Springer Science & Business Media, 2013.
- [5] G. Alvarez, “Metodología de Identificación Difusa Basada en el Estudio de Controlabilidad de Sistemas Dinámicos,” p. 172.
- [6] I. Škrjanc, “Confidence interval of fuzzy models: An example using a waste-water treatment plant,” *Chemometrics and Intelligent Laboratory Systems*, vol. 96, no. 2, pp. 182–187, Apr. 2009. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0169743909000124>
- [7] —, “Fuzzy confidence interval for pH titration curve,” *Applied Mathematical Modelling*, vol. 35, no. 8, pp. 4083–4090, Aug. 2011. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0307904X11000904>
- [8] Matlab, “trainlm, levenberg-marquardt backpropagation.” [Online]. Available: <https://www.mathworks.com/help/deeplearning/ref/trainlm.html>
- [9] M. Nørgaard, O. Ravn, N. K. Poulsen, and L. K. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems*, ser. Advanced Textbooks in Control and Signal Processing, M. J. Grimble and M. A. Johnson, Eds. London: Springer London, 2000. [Online]. Available: <http://link.springer.com/10.1007/978-1-4471-0453-7>
- [10] N. Cruz, L. G. Marin, and D. Saez, “Neural Network Prediction Interval Based on Joint Supervision,” in *2018 International Joint Conference on Neural Networks (IJCNN)*. Rio de Janeiro: IEEE, Jul. 2018, pp. 1–8. [Online]. Available: <https://ieeexplore.ieee.org/document/8489264/>
- [11] “Multi-Task Learning in Tensorflow: Part 1,” library Catalog: www.kdnuggets.com. [Online]. Available: <https://www.kdnuggets.com/multi-task-learning-in-tensorflow-part-1.html/2/>