

# Hibridación en Modelado Difuso y Minería de Datos

Mercedes Valdés, Antonio F. Gómez-Skarmeta y Juan A. Botía

Departamento de Ingeniería de la Información y las Comunicaciones, Universidad de Murcia. Campus de Espinardo. E-30071. Murcia, Spain.  
{mvaldes, skarmeta, juanbot}@dif.um.es

**Resumen** El *Modelado Difuso* es un enfoque efectivo para la generación de modelos de sistemas basado en la lógica y conjuntos difusos y que describe el comportamiento de los sistemas en base a reglas *SI – ENTONCES*. El *Modelado Difuso Conducido por Datos* (DDFM) intenta generar modelos difusos a partir de un conjunto de ejemplos entrada-salida. Se identifican varias fases a la hora de llevar a cabo el proceso de DDFM. Básicamente, se pueden resumir en: la *detección del número de reglas*, la *generación de las propias reglas* y el *ajuste de sus parámetros*. Nuestras investigaciones, ponen de manifiesto la validez y éxito de las estrategias que abordan el problema combinando varias técnicas, cada una de ellas resolviendo una o varias de estas fases. En este artículo aplicamos hibridaciones de varios algoritmos de clustering y redes neuronales para resolver distintos problemas de regresión pertenecientes a diferentes dominios de aplicación mediante la obtención de sistemas de reglas difusas a partir de los datos.

## 1. Introducción

Las empresas e instituciones se enfrentan día a día con grandes cantidades de datos. De hecho, los datos son la muestra evidente de la Revolución de la Información. *Vivimos en una sociedad rica en datos y pobre en conocimiento* [10]. Por tanto, hay que darle a los datos la importancia que se merecen, al menos, tras la aparición de las tecnologías que se pueden englobar en la extensa área del Tratamiento Inteligente de Información. (Minería de Datos, Descubrimiento del Conocimiento, Identificación de Sistemas, etc.). De un modo general, podemos decir que todas ellas tratan de *dar sentido a los datos*. De hecho, todas estas tecnologías empiezan y terminan en los datos: se necesitan los datos para extraer conocimiento de ellos y construir modelos, se necesitan datos para validar dichos modelos, el resultado de esta extracción no puede ser mejor que los datos de los que partió, el modelo construido infiere datos.

Uno de los enfoques más utilizados en los últimos tiempos para la obtención de modelos a partir de datos entrada-salida es el **Modelado Difuso Conducido por Datos** (o **DDFM**, del inglés *data-driven fuzzy modeling*) cuyo objetivo es extraer de ellos conocimiento representado en forma de **sistema de inferencia difuso** (FIS). Un FIS es una herramienta computacional para describir sistemas,

que estructura el conocimiento en base a reglas *SI-ENTONCES* formadas por conjuntos difusos que representan conceptos vagos e imprecisos. Se fundamenta en la teoría de conjuntos y razonamiento difusos [12].

El proceso de DDFM consta fundamentalmente de tres fases: la *detección del número de reglas* del que consta el modelo, la *generación de las propias reglas SI-ENTONCES* y el *ajuste de los parámetros* de los conjuntos difusos que intervienen en ellas. Esta naturaleza modular favorece la utilización de combinaciones de técnicas para llevar a cabo el proceso, cada una de ellas resolviendo una de sus fases. En este capítulo se pone de manifiesto la validez y éxito de este enfoque híbrido de DDFM mediante la aplicación de este tipo de estrategias compuestas a distintos problemas de regresión procedentes de diferentes dominios.

El resto del capítulo se estructura del siguiente modo: en la sección 2 describimos más profundamente los sistemas de inferencia difusos y más concretamente los de tipo TSK que son los que se van a generar a partir de las técnicas de DDFM que aquí utilizamos. La sección 3 está dedicada a la estrategia híbrida de DDFM que vamos a aplicar describiendo cada una de sus componentes, entre las cuales encontramos algoritmos de detección de agrupamientos o *clustering* y una *red neuro-difusa* [13]. En la sección 4 describimos los resultados de la aplicación de la estrategia híbrida de DDFM a distintos problemas de diferentes dominios. Finalmente, en la sección 5 se exponen las conclusiones del trabajo.

## 2. Modelo de Representación: Modelo Difuso de Tipo TSK

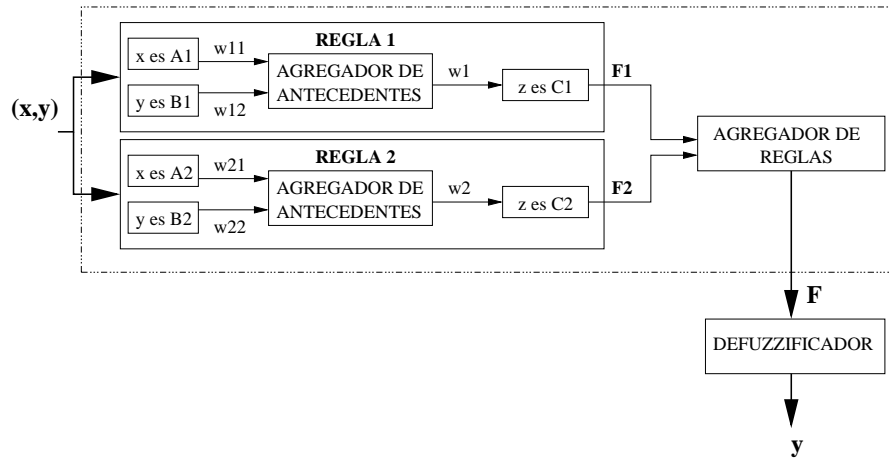
Se pueden distinguir tres unidades conceptuales dentro de un FIS: la **base de reglas**, un **diccionario** en el que se definen los conjuntos difusos que aparecen en la base de reglas (**funciones de pertenencia**) y por último, un **mecanismo de razonamiento difuso** (FRM, del inglés, *fuzzy reasoning method*) mediante el cual el sistema infiere salidas a partir de entradas. En la figura 1 podemos ver el diagrama de bloques del proceso de inferencia de un FIS, cuya base de reglas es

Regla 1: SI  $x$  es  $A_1$  Y  $y$  es  $B_1$  ENTONCES  $z$  es  $C_1$

ADEMAS

Regla 2: SI  $x$  es  $A_2$  Y  $y$  es  $B_2$  ENTONCES  $z$  es  $C_2$ ,

donde  $x$  e  $y$  son las variables de entrada,  $z$  es la variable de salida y  $A_i$ ,  $B_i$  y  $C_i$  con  $i = 1, 2$ , son conjuntos difusos. La entrada al sistema puede ser difusa o real (*crisp*). Para obtener la salida debida a esa entrada, el sistema ha de determinar el **grado de cumplimiento o disparo** de cada una de las proposiciones difusas atómicas de los antecedentes (por ejemplo,  $x$  es  $A_1$ , es una proposición difusa atómica). En el siguiente paso, el sistema calcula el grado de cumplimiento de la proposición difusa compuesta que forma el antecedente de cada regla, agregando las proposiciones atómicas mediante la conectiva Y. También se tienen que realizar las implicaciones *SI-ENTONCES* y la agregación de las reglas debida a la conectiva *ADEMÁS*. La salida total del sistema es un valor difuso. Sin



**Figura 1.** Diagrama de bloques de un sistema de inferencia difuso

embargo, en muchas ocasiones es requerido un valor *crisp*. Para ello, ocurre un postproceso de *defuzzificación* en el que se obtiene un valor *crisp* representativo del conjunto difuso de salida.

En los modelos TSK [21,20] los consecuentes de las reglas son funciones polinomiales de las variables de entrada:

$$\begin{aligned} &\text{SI } x_1 \text{ es } A_1 \text{ Y } \dots \text{ Y } x_p \text{ es } A_p \\ &\text{ENTONCES } y_1 = a_1(x_1, \dots, x_p) \text{ Y } \dots \text{ Y } y_h = a_h(x_1, \dots, x_p) \end{aligned}$$

donde cada variable de salida  $y_k$  con  $k = 1, \dots, h$ , viene dada por una función polinomial  $a_k$  definida sobre las variables de entrada  $x_1, \dots, x_p$ . Con este tipo de consecuentes se pretende que cada regla describa un comportamiento local del sistema. Geométricamente, las reglas del modelo TSK corresponden a una aproximación de la función original mediante una combinación de funciones lineales.

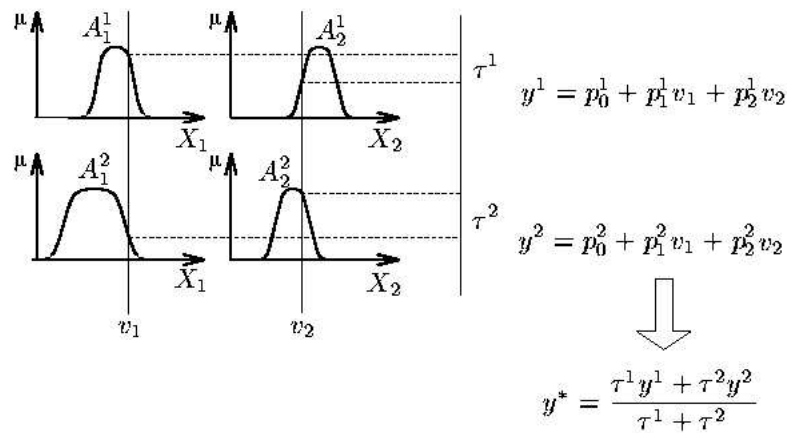
Supongamos, sin pérdida de generalidad, que  $h = 1$  (es decir, consideramos una única variable de salida). Para obtener la salida total inferida por un modelo TSK (ya sea de orden 0 o de primer orden), en primer lugar hay que calcular el grado de disparo de cada regla, al que denotaremos como  $\tau_i$ , con  $i = 1, \dots, r$  y siendo  $r$  el número de reglas del modelo. Dada la entrada  $x_1, \dots, x_p$ , hay que realizar la agregación de las proposiciones atómicas del antecedente de la regla  $i$ -ésima mediante el operador  $Y$ , que se calcula como:

$$\tau_i = A_{i1}(x_1) \cdot A_{i2}(x_2) \cdot \dots \cdot A_{ip}(x_p),$$

Puede decirse que el resto del proceso de inferencia se realiza en un sólo paso, de modo que la salida  $y$  se obtiene de la siguiente forma:

$$y = \frac{\sum_{i=1}^r \tau_i y_i}{\sum_{i=1}^r \tau_i}$$

siendo  $y_i$  el consecuente de la regla  $i$ -ésima. Puesto que cada regla tiene salida *crisp*, no se requiere un tiempo extra para la defuzzificación. La figura 2 muestra gráficamente el funcionamiento del modelo de inferencia TSK para dos entradas y dos reglas.



**Figura 2.** Modelo de inferencia TSK con 2 entradas y 2 reglas.

En este capítulo vamos a obtener modelos difusos de tipo TSK a partir de las técnicas de DDFM que expondremos en la siguiente sección.

### 3. Un Enfoque híbrido para DDFM

Una vez que hemos introducido cuál es el modelo de representación de conocimiento que vamos a utilizar, en esta sección exponemos los algoritmos que vamos a aplicar para extraer dicho conocimiento. El enfoque que utilizamos para el DDFM es híbrido en el sentido de que se aplican diferentes técnicas para abordar las fases de las que consta esta tarea. En cuanto a la parte de identificación de las reglas, es habitual aplicar técnicas de clustering, detectado agrupamientos en los datos, cada uno de los cuales será una potencial regla difusa. Los centroides

de estos agrupamientos se proyectan en cada una de las dimensiones (atributos) de entrada obteniéndose así los parámetros iniciales de los conjuntos difusos de los antecedentes de las reglas. Después, tales parámetros se pueden optimizar mediante las capacidades adaptativas proporcionadas por una red neuronal.

### 3.1. Detección del Número de Reglas Mediante Métodos de Clustering Auto-Organizativos

Nos referimos con el adjetivo de *auto-organizativos* a aquellos métodos de clustering capaces de detectar por sí mismos el número óptimo de agrupaciones subyacentes en los datos. La ventaja de la utilización de métodos de clustering auto-organizativos es que, además de resolver el problema de la identificación del número de reglas, generan centroides que pueden utilizarse como valores iniciales para algún método posterior de generación de reglas como, por ejemplo, otros métodos de clustering cuya bondad suele depender mucho de la partición inicial.

**Método de Clustering de la Montaña (MCM).** Yager y Filev propusieron en [23] un método de clustering simple y efectivo. Se basa en la división del espacio de los datos en una cuadrícula. Los vértices de ésta se consideran posibles clusters. Para determinar cuáles de esos candidatos pasan a ser centroides se basan en una medida de potencial o densidad de datos llamada *función montaña*. De ahí su nombre, *Mountain Clustering Method* (MCM). La función montaña para el punto  $v_i$  se define como:

$$M(v_i) = \sum_{k=1}^n e^{-\frac{\|z_k - v_i\|^2}{2\sigma^2}} \quad (1)$$

donde  $\sigma$  afecta a la altura y a la suavidad de la función montaña. Cada dato  $z_k$  contribuye a la función montaña del vértice  $v_i$ , siendo esta contribución inversamente proporcional a la distancia entre  $z_k$  y  $v_i$ . De modo que un vértice con muchos puntos cercanos tiene mayor valor para la función montaña que uno con pocos datos cerca.

El proceso de identificación de los clusters es el siguiente: en primer lugar, se calcula la función montaña para cada uno de los vértices. El siguiente paso es seleccionar como cluster aquel con mayor valor para ésta. Si existen varios puntos con el mismo potencial, entonces se escoge uno de ellos al azar. Para seleccionar el  $j$ -ésimo cluster, es necesario eliminar el efecto de los detectados anteriormente. Es decir, queremos evitar que se escojan puntos cercanos al último cluster detectado. Para ello se reduce la función montaña de cada vértice mediante la fórmula:

$$M_j(v_i) = M_{j-1}(v_i) - M_{j-1}^* e^{-\frac{\|z_{j-1}^* - v_i\|^2}{2\beta^2}}$$

donde  $M_j(v_i)$  representa la nueva función montaña para  $v_i$ ,  $M_{j-1}(v_i)$  es la función montaña de  $v_i$  anterior,  $z_{j-1}^*$  es el último centroide detectado,  $M_{j-1}^*$  es su correspondiente función montaña y  $\beta$  es una constante positiva.

Una vez que se tienen los valores  $M_j(v_i)$  para cada vértice  $v_i$  se escoge el siguiente cluster de la misma manera. El proceso continúa hasta que la función montaña máxima actual  $M_{j-1}^*$  es muy pequeña en comparación con la función montaña máxima original  $M_1^*$ :

$$\frac{M_{j-1}^*}{M_1^*} < \delta \quad (2)$$

Las principales características de este algoritmo son las siguientes:

- Es relativamente robusto frente al ruido. Puesto que basa la elección de un punto como centroide en la medida de la densidad de puntos cercanos a él, entonces, si un punto es ruidoso tendrá un potencial pequeño y por tanto menos posibilidades de ser seleccionado como cluster.
- La bondad de la partición generada depende en gran medida del tamaño de las particiones en el espacio de definición de los datos. Cuanto más pequeñas son las particiones, mejores son los resultados, sin embargo, es computacionalmente más costoso. También aumenta el coste computacional con la dimensión de los datos siendo este aumento exponencial. Por ejemplo, si tenemos un problema de clustering con 5 variables, y decidimos dividir cada una de las dimensiones del espacio de datos en 10 partes, esto resultará en  $10^5$  vértices para los que hay que calcular una y otra vez su función montaña.

Para solucionar los problemas del elevado coste computacional debido al aumento de la dimensión del problema, se propuso una modificación de este método que exponemos a continuación.

**Método de Clustering Substractivo (SCM).** En [6], Chiu propone un algoritmo basado en el MCM pero en el que se trata de reducir su coste computacional. Se trata del *método de clustering substractivo* (SCM). En primer lugar hace que la complejidad sea independiente de la dimensión del problema. Además, elimina la necesidad de decidir el tamaño de las particiones. Por último, se basa en un criterio de parada más sofisticado que el MCM.

La estrategia consiste en considerar los propios puntos del conjunto de datos como los candidatos a clusters. De forma que el coste computacional deja de depender de la dimensión del problema y pasa a depender únicamente del tamaño del conjunto.

La función montaña para el dato  $z_i$  se obtiene según la fórmula:

$$M(z_i) = \sum_{k=1}^n e^{-\frac{\|z_k - z_i\|^2}{r_a^2}}$$

donde  $r_a$  es un constante positiva que determina un radio de vecindad. Los puntos  $z_k$  fuera de ese radio tienen poca influencia en el valor de la función montaña de  $z_i$ , de manera que tal influencia se reduce exponencialmente con el cuadrado de la distancia entre  $z_k$  y  $z_i$ .

Del mismo modo, una vez seleccionado un cluster, hay que revisar el potencial de cada  $x_i$  para eliminar su efecto:

$$M_j(z_i) = M_{j-1}(z_i) - M_{j-1}^* e^{-\frac{\|z_{j-1}^* - z_i\|}{(\frac{r_b}{2})^2}}$$

donde  $r_b$  es otro radio de vecindad de modo que los puntos situados a una distancia de  $z_{j-1}^*$  menor que  $r_b$  van a tener mayor reducciones de potencial, evitándose así escoger clusters demasiado cercanos.

La última modificación del método SCM con respecto al MCM es un criterio de terminación más sofisticado. Puesto que la elección del umbral  $\delta$  de la ecuación 2 tiene gran influencia sobre los resultados y es de difícil elección, en [6], se escoge un intervalo  $[\epsilon_{min}, \epsilon_{max}]$ . Los datos con valores de montaña por encima de  $\epsilon_{max}$  se seleccionan como clusters. Los datos con potencial menor que  $\epsilon_{min}$  se rechazan definitivamente. Para los puntos en el rango  $[\epsilon_{min}, \epsilon_{max}]$ , se estudia si existe compromiso entre tener un potencial razonable y estar a una distancia suficientemente grande de los clusters existentes.

Para terminar, diremos que este método mejora al MCM en su tiempo computacional y no depende de la dimensionalidad de los datos sino del tamaño del conjunto. También aporta un criterio de terminación más sofisticado. Sin embargo, es algo más sensitivo al ruido al existir la posibilidad de escoger como candidato un punto que puede ser ruidoso. Además un problema inherente al SCM es que dará un buen resultado si ocurre que los clusters “reales” o “buenos” coinciden o están cerca de algunos de los puntos de datos.

### 3.2. Métodos de Clustering para la Generación de Reglas

Las técnicas de clustering expuestas en la sección anterior escogen los centroides entre los vértices de la cuadrícula en el caso del MCM o entre los ejemplos del conjunto de entrenamiento en el caso del SCM. Sin embargo, los centroides “reales” no tienen por qué coincidir con tales puntos y por tanto son necesarios mecanismos de optimización de los agrupamientos. En esta sección exponemos dos métodos de clustering que optimizan un conjunto de agrupamientos inicial dando lugar a otro que puede ser una primera aproximación al modelo difuso que es el objetivo final de la estrategia híbrida.

**Clustering HCM (*Hard C-Means*).** Dados un conjunto de puntos  $Z = \{z_1, \dots, z_n\}$  y un número preestablecido  $c$ , este método, también conocido como de las *k-medias*, obtiene un conjunto de  $C$  agrupamientos  $C = \{C_1, \dots, C_c\}$ , y encuentra los prototipos  $V = \{v_1, \dots, v_c\}$  para dichos clusters de forma que se minimice una función de costo  $J_1$ , definida como:

$$J_1(U, V : Z) = \sum_{i=1}^c \sum_{z_k \in C_i} D_{ik}^2 \quad (3)$$

donde

$$D_{ik}^2 = \|z_k - v_i\|^2 \quad (4)$$

Se trata de una de las funciones más utilizadas por los algoritmos de clustering basados en función objetivo y representa la suma de los errores cuadráticos dentro de un grupo.

La matriz que define la partición es  $U$  tal que  $U = [\mu_{ik}]$ , donde  $\mu_{ik} = 1 \Leftrightarrow z_k \in C_i$ . Si los  $v_i$  están fijos, los  $\mu_{ik}$  que minimizan  $J_1$  son:

$$\mu_{ik} = \begin{cases} 1 & \text{si } D_{ik}^2 \leq D_{ij}^2, \forall j \neq i \\ 0 & \text{en otro caso.} \end{cases} \quad (5)$$

Por otro lado, si los  $\mu_{ik}$  están fijos, los centroides óptimos  $v_i$  que minimizan  $J_1$  son el promedio de los puntos de cada grupo  $C_i$ :

$$v_i = \frac{1}{|C_i|} \sum_{z_k \in C_i} z_k \quad (6)$$

Las ecuaciones 4, 5 y 6 son la base del siguiente algoritmo iterativo:

**Inicialización:** Dado el conjunto de datos  $Z = \{z_1, \dots, z_n\}$ , fijar  $c$ ,  $T$ ,  $\| \|_A$  y  $\epsilon > 0$ . Hacer  $t = 0$ . Inicializar aleatoriamente la matriz  $U_t$ .

**Repetir** para  $t = 1, \dots, T$

1. **Cálculo de los prototipos**  $V^t$  en base a la ecuación 6.
2. **Cálculo de las distancias** en base a la ecuación 4.
3. **Actualizar**  $U^t$  en base a la ecuación 5.

**Hasta**  $\|U^t - U^{t-1}\| < \epsilon$ .

**Clustering FCM (Fuzzy C-Means).** La función  $J_1$  presentada en la ecuación 3 ha sido posteriormente modificada por muchos autores, siendo la formulación más general:

$$J_m(U, V : Z) = \sum_{i=1}^c \sum_{k=1}^n (\mu_{ik})^m D_{ikA}^2 \quad (7)$$

Esta función generaliza la anterior en varios sentidos. En primer lugar la matriz  $U$  está formada por elementos del intervalo  $[0, 1]$ , de modo que  $J_m$  representa el costo de una partición difusa de los datos de  $Z$  (cada dato pertenece a todos los clusters con mayor o menor grado). En segundo lugar,  $m \in [1, \infty)$  indica un exponente de peso que determina el grado de borrosidad (*fuzziness*) de los clusters. En  $J_1$  suponíamos  $m = 1$ . En tercer lugar

$$D_{ikA} = \|z_k - v_i\|_A = \sqrt{(z_k - v_i)^T A (z_k - v_i)} \quad (8)$$

es la distancia ortogonal entre  $z_k$  y  $v_i$  en la norma  $A$  siendo  $A$  cualquier matriz  $p \times p$  positiva. En el caso de  $J_1$  era  $A = I$  (matriz identidad) que define la distancia euclídea entre  $v_i$  y  $z_k$ , pero como veremos, otras elecciones de  $A$  inducen otros tipos de distancias.



La función  $J_m$  alcanza su valor mínimo para:

$$v_i = \frac{\sum_{k=1}^n \mu_{ik}^m z_k}{\sum_{k=1}^n \mu_{ik}^m} \quad (9)$$

siendo

$$u_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{D_{ikA}}{D_{jkA}}\right)^{\frac{2}{m-1}}} \quad (10)$$

El algoritmo más utilizado para minimizar 7 es el que Bezdek propone en [3]:

**Inicialización:** Dado el conjunto de datos  $Z = \{z_1, \dots, z_n\}$ , fijar  $c$ ,  $T$  y  $\epsilon > 0$ .

Hacer  $t = 0$ . Inicializar aleatoriamente la matriz  $U_t$ .

**Repetir** para  $t = 1, \dots, T$

1. **Cálculo de los prototipos**  $V^t$  en base a la ecuación 9.
2. **Cálculo de las distancias** en base a la ecuación 8.
3. **Actualizar**  $U^t$  en base a la ecuación 10.

**Hasta**  $\|U^t - U^{t-1}\| < \epsilon$ .

Tenemos que recalcar que cuando  $D_{skA} = 0$  para uno o más centroides  $v_s$  con  $s \in S \subset \{1, 2, \dots, c\}$  los correspondientes  $\mu_{sk}$  no se pueden calcular según la ecuación 10. En estos casos se hace  $\mu_{jk} = 0$  para  $j \in \bar{S}$  y el resto de pertenencias se asigna arbitrariamente entre los  $\mu_{sk}$  de modo que  $\sum_{s \in S} \mu_{sk} = 1$ .

Veamos el significado así como algunas consideraciones sobre los parámetros manejados en el FCM:

- **Parámetro de terminación** ( $\epsilon$ ):  $\epsilon$  es un umbral, que determina el criterio de terminación. Los valores que usualmente se dan a  $\epsilon$  son 0.001 y 0.01 [1].
- **Parámetro de fuzziness de la partición** ( $m$ ): Cuanto mayor es  $m$  más difusos son los clusters y cuanto más cercano es  $m$  a su límite inferior (1), más *crisp* son las agrupaciones. Con esto queremos decir que la influencia que ejercen sobre un dato  $z_k$  centroides que están alejados de él, será mayor cuanto mayor sea  $m$ . Por el contrario,  $z_k$  tendrá grados de pertenencia a esos clusters más cercanos a 0 cuanto más próximo a 1 sea  $m$ .

Los estudios acerca de  $m$  llevan a que no existe ninguna razón teórica para elegir un valor particular de  $m$  [19], si no que es una cuestión dependiente de la complejidad de los datos [8]. Un estudio hecho por Pal y Bezdek en su trabajo [18] determina que la mejor elección para  $m$  oscila entre 1.5 y 2.5, siendo el punto medio 2 el valor más utilizado por los usuarios del FCM. En un conjunto de datos muy próximos, la tendencia para un valor de  $m$  cercano a 2 será un alto grado de solapamiento, y esto no es conveniente si nuestro interés es obtener clusters distinguibles, para lo cual elegiríamos mejor un valor cercano a 1.5. Lo que sí revelan los estudios es que la convergencia FCM no se ve afectada por el valor de  $m$  [19].

- **Matriz inicial ( $U^0$ ):** La elección de la matriz inicial  $U^0$  si afecta considerablemente al resultado de los algoritmos FCM y HCM, al basarse en estrategias de tipo ascenso de colinas en los que diferentes inicializaciones llevan a distintos óptimos locales. Para asegurarnos de obtener una partición “buena”, será necesario obtener unos centroides iniciales mediante algoritmos de clustering auto-organizativos y a partir de ellos aplicar FCM para optimizar los centroides.
- **Número de clusters ( $c$ ):** Este parámetro es de los más importantes porque influye mucho en la obtención de una partición que satisfaga al usuario. Una de las bondades de las técnicas de análisis cluster es que no imponen a los datos una estructura *a priori*, si no que detectan la que en ellos subyace. Sin embargo, la necesidad que tienen estos algoritmos de la familia FCM de que se les proporcione el número concreto de agrupaciones que deben encontrar, impone una estructura a esos datos con la que, probablemente, no se corresponden. De nuevo, este problema se puede solucionar aplicando previamente una de las técnicas expuestas en la sección 3.1.
- **Norma que induce la medida de distancia ( $A$ ):** La importancia de la matriz  $A_{p \times p}$  radica en que determina qué tipo de distancia se utiliza para medir la similitud entre los datos de  $Z$ . Las tres medidas de distancia más utilizadas son:
  1. La distancia euclídea, que se obtiene al utilizar como norma la matriz identidad ( $A = I$ ):

$$D_{ikA}^2 = \|z_k - v_i\|^2 = \sqrt{\sum_{j=1}^p (z_{kj} - v_{ij})^2}$$

2. La distancia diagonal, que se obtiene al utilizar como norma la inversa de la matriz diagonal de las varianzas muestrales del conjunto de datos  $Z$  ( $A = \text{diag}(Z)^{-1}$ ):

$$D_{ikA}^2 = \|z_k - v_i\|_D^2 = \sqrt{(z_k - v_i)^T [\text{diag}(Z)]^{-1} (z_k - v_i)}$$

3. La distancia de Mahalanobis, obtenida al utilizar la inversa de la matriz de covarianzas muestrales de  $Z$  ( $A = \text{cov}(Z)^{-1}$ ):

$$D_{ikA}^2 = \|z_k - v_i\|_M^2 = \sqrt{(z_k - v_i)^T [\text{cov}(Z)]^{-1} (z_k - v_i)} \quad (11)$$

El tipo de medida influye en la forma de los clusters. La distancia euclídea induce clusters cuyas curvas de nivel son hiperesferas. La distancia diagonal lleva a clusters hiperelipsoidales cuyos ejes son paralelos a los ejes de coordenadas. Por último, la distancia de Mahalanobis da lugar también a clusters hiperelipsoidales pero con cualquier orientación respecto a los ejes de coordenadas.

### 3.3. Optimización de Parámetros Mediante Redes Neuro-Difusas

Mediante la aplicación secuencial de uno de los mecanismos de clustering auto-organizativo y uno de los algoritmos de optimización de clusters se obtiene una primera aproximación a un modelo de reglas difusas. Basta con proyectar los centroides en las distintas dimensiones (atributos) de entrada obteniéndose los parámetros iniciales de los antecedentes de las reglas difusas. Es el momento de ajustarlos para generar el modelo difuso buscado.

Los métodos de aprendizaje de las redes neuronales capacitan a los sistemas difusos para aprender información nueva a partir de ejemplos de entrada-salida. Por su parte, los conjuntos difusos capacitan a las redes neuronales para presentar la información aprendida en una forma interpretable para los humanos [22]. Estas son las dos razones fundamentales por las que las herramientas que se nutren de estas dos áreas son un mecanismo eficiente a la hora de modelar sistemas reales. Se trata de las **redes adaptativas neuro-difusas** [7,13,12].

Concretamente, en este trabajo aplicamos la red neuronal ANFIS (*Adaptive Neuro-based Fuzzy Inference System*) propuesta por Jang [13,12] y que es funcionalmente equivalente a un sistema TSK. Su arquitectura se muestra en la figura 3.

Los nodos situados en la misma capa realizan funciones similares. Los nodos representados con rectángulos son adaptativos, mientras que los nodos representados por círculos no lo son:

- Capa 0: corresponde a las entradas  $x_1, \dots, x_p$ .
- Capa 1: los nodos de esta capa son adaptativos, y cada uno calcula el grado de pertenencia  $\mu_{ij}$  de la entrada  $j$ -ésima al conjunto difuso  $A_{ij}$  que viene dado por una función *bell-shaped*:

$$A_{ij} = \frac{1}{1 + \left| \frac{x_j - c_j^i}{a_j^i} \right|^{2b_j^i}} \quad i = 1, \dots, c, \quad j = 1, \dots, p,$$

donde  $x_i$  es la entrada y  $\{a_j^i, b_j^i, c_j^i\}$  es el conjunto de parámetros.

- Capa 2: los nodos de esta capa son no adaptativos y se define la salida del nodo  $i$ -ésimo como el producto de sus entradas:

$$\beta_i = \prod_{j=1}^p \mu_{ij} \quad i = 1, \dots, r,$$

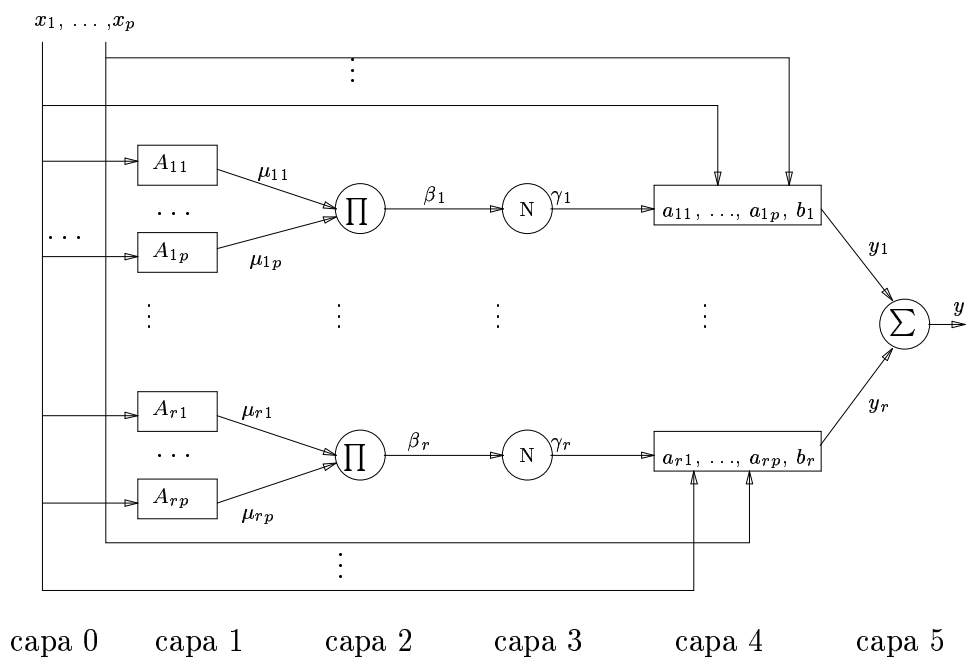
donde cada salida  $\beta_i$  corresponde al grado de disparo de la  $i$ -ésima regla.

- Capa 3: los nodos de esta capa son no adaptativos. El nodo  $i$ -ésimo obtiene el grado de disparo normalizado de la regla  $i$ -ésima

$$\gamma_i = \frac{\beta_i}{\sum_{k=1}^r \beta_k}$$

- Capa 4: cada nodo de esta capa es adaptativo y los parámetros del nodo  $i$ -ésimo son  $a_{i1}, \dots, a_{ip}, b_i$ . Su salida se corresponde con la salida parcial de la  $i$ -ésima regla,

$$y_i = \gamma_i (a_{i1}x_1 \dots + a_{ip}x_p + b_i)$$



**Figura 3.** Arquitectura ANFIS.

- Capa 5: el nodo de esta capa es no adaptativo y su salida se define como la suma de las salidas parciales  $y_i$ :

$$y = \sum_{i=1}^r y_i$$

Los nodos de la capa 1 y los de la capa 4 se corresponden respectivamente con los parámetros de los conjuntos difusos de los antecedentes de un sistema TSK y los coeficientes de las funciones lineales de sus consecuentes. Así que, el entrenamiento de la red ANFIS constituye la realización de una fase de ajuste de parámetros sobre el modelo inicial representado en la arquitectura. Es decir, el punto de partida es el modelo TSK inicial dado por la arquitectura.

Se puede observar en la figura 3 que, si los parámetros de los antecedentes quedan fijos, la salida del sistema se puede expresar como una combinación lineal de los parámetros de los consecuentes. La salida  $y$  puede ser escrita entonces de la siguiente manera:

$$\begin{aligned} y = & \gamma_1(a_{11}x_1 + \dots + a_{1p}x_p + b_1) \\ & + \dots + \\ & \gamma_r(a_{r1}x_1 + \dots + a_{rp}x_p + b_r) \end{aligned} \quad (12)$$

y esta ecuación puede reescribirse como:

$$\begin{aligned} f = & \gamma_1x_1a_{11} + \dots + \gamma_1x_pa_{1p} + \gamma_1b_1 \\ & + \dots + \\ & \gamma_rx_1a_{r1} + \dots + \gamma_rx_pa_{rp} + \gamma_rb_r \end{aligned} \quad (13)$$

que es lineal en los parámetros de los consecuentes  $a_{i1}, \dots, a_{ip}, b_i$  con  $i = 1, \dots, p$ . Por tanto, el método de ajuste por mínimos cuadrados se puede utilizar para optimizar los coeficientes lineales de los consecuentes.

Se puede entonces combinar el método de *backpropagation* con el método de cuadrados mínimos para modificar los parámetros de la red [12]. Para ello, cada época está compuesta por una pasada hacia adelante y una pasada hacia atrás. En la pasada hacia adelante, para cada vector de entrada, se evalúa la red hasta la capa 4, y los parámetros de los consecuentes son identificados mediante el método de cuadrados mínimos. A continuación, se calculan los errores para cada par del conjunto de entrenamiento y, en la pasada hacia atrás se propagan las señales del error y los parámetros de las premisas son modificados por el mecanismo clásico de *backpropagation*.

## 4. Aplicaciones

Una vez introducida la estrategia híbrida para la obtención de modelos de reglas difusas, en esta sección exponemos varios ejemplos de aplicación. En una primera parte aplicamos DDFM al problema de la predicción de variables climáticas a partir de los datos recogidos a través de una red de estaciones agroclimáticas y almacenados en una gran base de datos.

Posteriormente, utilizamos nuestras técnicas para la generación de un modelo difuso que resuelve un problema típico en la navegación de robots móviles: la fusión de las medidas de distancia procedentes de distintos sensores con la intención de obtener una única medida más parecida a la real.

#### 4.1. DDFM Híbrido para la Predicción de Variables Agroclimáticas

En esta sección vamos a exponer varios ejemplos de aplicación del DDFM para la predicción de variables agroclimáticas como son la evapotranspiración de referencia y la radiación solar.

**La Evapotranspiración de Referencia ( $ET_0$ ).** El problema de la estimación de las necesidades hídricas de un cultivo es de gran importancia en zonas donde este recurso escasea, como ocurre por ejemplo en el sureste español. La predicción de dichas necesidades se traduce a la estimación de cierta magnitud llamada *evapotranspiración de referencia* ( $ET_0$ ). La  $ET_0$  es la cantidad de agua que la planta pierde por causa de la evaporación y la transpiración, y por tanto, es la cantidad de agua que se ha de reponer. Su valor depende de las condiciones climáticas particulares y del tipo de cultivo. Una vez que se ha obtenido la  $ET_0$ , se calcula la evapotranspiración del cultivo ( $ET_c$ ) por medio de la ecuación  $ET_c = K_c \times ET_0$ , donde  $K_c$  es una constante correctora para adaptar la evapotranspiración a una planta particular, en una determinada fase de crecimiento, etc.. Por último, la cantidad de agua ( $N_t$ ) que el cultivo necesita se obtiene de la diferencia de  $ET_c$  y la lluvia efectiva ( $P_e$ ), que es la cantidad de agua de lluvia útil para el cultivo. Así, la expresión final es  $N_t = ET_c - P_e$ .

**El Modelo Evaporimétrico de Cubeta Clase A.** Tradicionalmente, se ha estimado el valor de la  $ET_0$  a partir de datos climáticos y utilizando modelos matemáticos definidos analíticamente. Uno de los más usados es el llamado *Modelo Evaporimétrico de Cubeta Clase A* (MECCA) [16]. Dicho modelo se basa en la observación de los cambios del nivel de agua en una cubeta y en ciertas variables climáticas que influyen en el consumo de agua por parte de la plantación: la velocidad del viento y la humedad del ambiente. La expresión que relaciona la  $ET_0$  con estas tres variables es:

$$ET_0 = K_p \times E_0$$

siendo  $E_0$  la evaporación del agua de la cubeta dada en milímetros/día y  $K_p$  el llamado coeficiente corrector del tanque, que se calcula mediante la siguiente expresión:

$$K_p = a_0 + a_1U + a_2H_r + a_3d + a_4H_r^2 + a_5d^2 + a_6UH_r^2 + a_7dH_r^2$$

donde  $U$  es la velocidad media del viento,  $H_r$  es la humedad relativa media,  $d$  es la distancia a barlovento de la cubeta a la cubierta vegetal y  $a_i$  con  $0 \leq i \leq 7$  son coeficientes obtenidos en una situación ideal. Sin embargo, tales condiciones

ideales se pueden reproducir en el laboratorio pero no suelen darse fuera de él. Es por ello por lo que los expertos necesitan adaptar los coeficientes  $a_i$  a las condiciones de las zonas particulares en las que se pretende medir la  $ET_0$ . No obstante, este ajuste no es una tarea sencilla y a menudo se basa en experiencia subjetiva y en procesos de ensayo-error.

**DDFM para la Predicción de la  $ET_0$ .** La Red Climática de la Región de Murcia está compuesta por 64 estaciones que disponen de diferentes tipos de sensores para medir variables climáticas. En este sistema de información, la determinación de la  $ET_0$  en base al modelo MECCA es una práctica bastante extendida. Esto no se debe a la bondad de dicho modelo, si no a que se basa en variables medibles a través de sensores de bajo coste que se encuentran en la mayoría de las estaciones climáticas de la red. Por esta razón, resulta interesante la aplicación de técnicas inteligentes y en concreto de estrategias de DDFM, no tanto para obtener los coeficientes  $a_i$  del modelo MECCA, como para generar otros modelos diferentes pero basados en las mismas variables climáticas, ya que éstas pueden ser medidas en la mayoría de las estaciones.

Podemos considerar que el modelo MECCA es una función  $f_{MECCA}$  tal que:

$$f_{MECCA} : U \times H_r \times E_0 \times d \rightarrow ET_0$$

Así que vamos a aplicar técnicas de DDFM para obtener un modelo difuso que aproxime  $f_{MECCA}$ . Para ello, disponemos de una base de datos procedentes de las 64 estaciones agroclimáticas. En ella existe una tabla llamada HORA que almacena series temporales horarias desde 1996 para diversas variables climáticas. Además existen dos tablas más almacenando respectivamente series diarias (tabla DIA) y semanales (tabla SEMANA) obtenidas agregando la información contenida en HORA. Los datos almacenados en las tablas HORA y DIA son demasiado ruidosos para utilizarlos en el proceso de entrenamiento. Así que hemos optado por utilizar 2580 tuplas de la forma  $\{U, H_r, E_0, d, ET_0\}$  procedentes de la tabla SEMANA correspondientes al periodo 1996-1999.

Hemos aplicado una modificación del mecanismo MCM (en el que los propios datos del conjunto de entrenamiento son los candidatos a centroides, nos referiremos a este método como MCM') para la detección del número de reglas, HCM para optimizar los centroides y ANFIS para ajustar los parámetros de las reglas.

Para evaluar la bondad de nuestros modelos, los cuales realizan la tarea de regresión, hemos utilizado la raíz cuadrada del error cuadrático medio (RMSE, *Root Mean Squared Error*) definido del siguiente modo:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{\sum_{i=1}^n (y_i^t - \hat{y}_i^t)^2}{n}} \quad (14)$$

siendo  $n$  el número de datos del conjunto de entrenamiento,  $y_i^t$  la salida deseada para el dato  $i$  y  $\hat{y}_i^t$  la salida inferida por el modelo.

En la tabla 1 podemos ver que los modelos difusos tienen mayor capacidad aproximativa que el modelo MECCA. Como vemos, hemos obtenido un modelo TSK de 12 reglas con un RMSE (ver ecuación 14) de 0.28. Este error es peor que el del modelo MECCA, que es de 1.69. Además, hemos entrenado con los mismos datos un perceptrón multicapa [17] (MLP) con 10 nodos en la capa oculta, obteniendo finalmente un RMSE de 0.34 que es mayor que el obtenido para el modelo TSK.

**Cuadro 1.** RMSE de distintos modelos para la  $ET_0$

Modelo	RMSE
MECCA	1.69
MLP (10 nodos ocultos)	0.34
MCM <sup>1</sup> +HCM+ANFIS (12 reglas)	0.28

**La Radiación Solar.** Como hemos dicho en la sección anterior, el interés por obtener modelos para la  $ET_0$  basados en el MECCA, no se debe a los buenos resultados de éste, sino a que se basa en la lectura de variables tales como la velocidad del viento, humedad, etc., que se pueden medir en la mayoría de las estaciones de la red climática. No obstante, existen otros modelos tradicionales que ofrecen mejores resultados que el MECCA, basados en otras variables climáticas diferentes. Uno de los más adecuados es el modelo de la *Radiación* que, como su propio nombre indica, necesita de la medida de la radiación solar para la obtención de la  $ET_0$ . Dicho modelo es actualmente uno de los métodos recomendados por la FAO (Organización Mundial para la Alimentación y la Agricultura) <sup>1</sup>. Por otro lado, el llamado modelo *FAO-Penman-Monteith*, es uno de los más extendidos en el sureste español por su adecuación a las condiciones particulares de esta zona. Este modelo también requiere, como una de sus entradas, la radiación solar. Sin embargo, el problema que existe con respecto al uso de estos dos modelos, es que la mayoría de las estaciones climáticas carece de la tecnología necesaria para medir la radiación solar. Por tanto, es nuestro interés y del de los expertos, la generación de modelos que, a partir de la radiación solar obtenida en estaciones preparadas para medirla, determinen la radiación en otros puntos en los que esto no sea posible. De esta manera, los dos modelos de cálculo de la  $ET_0$  referidos anteriormente podrán ser utilizados como está siendo recomendado por la FAO.

En las dos siguientes secciones presentamos dos soluciones para determinar la radiación solar en puntos donde no se dispone de la tecnología para medirla.

**Interpolación de la Radiación a partir de una Estación de Referencia.** Disponemos del mapa digital de elevación (*DEM*) de la Región de Murcia del

<sup>1</sup> <http://www.fao.org>



que podemos obtener las coordenadas  $(x, y, z)$  en el espacio  $X \times Y \times Z$  de todas las estaciones climáticas, donde  $X$  representa la longitud,  $Y$  representa la latitud y  $Z$  la altura sobre el nivel del mar. Vamos a referirnos al conjunto de las estaciones capaces de medir la radiación, o mejor dicho al conjunto de sus respectivas coordenadas en el  $DEM$ , como  $E_R$ . En la figura 4 podemos ver el mapa de la Región de Murcia y la distribución de las estaciones agroclimáticas, cada una de ellas representada por un código.



**Figura 4.** Situación de las estaciones de la Red Agroclimática de la Región de Murcia.

Pretendemos obtener un modelo difuso para inferir la radiación en un punto a partir de una entrada formada la radiación medida en otro punto del  $DEM$ , de la posición relativa de éste respecto al primero y del día del año. Nuestro problema se traduce en encontrar un modelo que aproxime cierta función:

$$f : [1, 365] \times R \times X \times Y \times Z \rightarrow R$$

De nuevo, vamos a utilizar el método MCM' para la determinación del número de reglas, HCM para identificar la estructura y ANFIS para ajustar los parámetros.

Para formar el conjunto de entrenamiento que guía el proceso de DDFM se utiliza la tabla SEMANA, ya que los datos diarios contienen muchos valores irregulares. Así que, en lugar de usar como entrada el día en el año utilizamos un número indicando la semana del año. Por tanto, cada tupla del conjunto de entrenamiento tiene la forma  $(w, r_i, x_{i0}, y_{i0}, z_{i0}, r_o)$ , donde  $w \in [1, 52]$  representa la semana,  $r_i$  y  $r_o$  son las radiaciones medidas respectivamente en dos estaciones

$e_i$  y  $e_o$ , siendo  $e_i, e_o \in E_R$  y  $e_i \neq e_o$ . La estación  $e_o$  se considera el lugar donde se pretende aproximar la radiación y la estación  $e_i$  se toma como referencia. Por tanto, el valor  $r_o$  es la salida deseada mientras que  $r_i$  es la radiación que se toma como entrada. La radiación se mide en vatios por metro cuadrado ( $W/m^2$ ) y suele tomar valores en el intervalo  $[0, 500] W/m^2$ . Por último,  $(x_{io}, y_{io}, z_{io}) = (|x_i - x_o|, |y_i - y_o|, |z_i - z_o|)$ , es el vector de coordenadas relativas de la estación  $e_o$  respecto a la estación  $e_i$ .

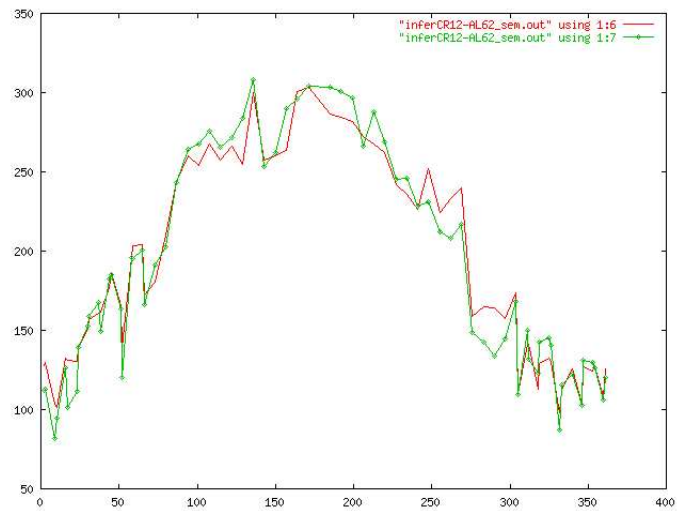
El conjunto de entrenamiento se compone de las 12578 tuplas generadas a partir de los datos de la tabla SEMANA. Hay que puntualizar, que no reservamos una parte de esto datos para la fase de validación porque el número elevado de variables de entrada de nuestro problema requiere un conjunto de datos lo suficientemente grande para garantizar un correcto entrenamiento. Finalmente, al llevar a cabo la estrategia de DDFM descrita anteriormente, se obtiene un modelo TSK formado por 37 reglas que alcanza un RMSE de entrenamiento de 15.53.

En este momento, tenemos un modelo capaz de inferir la radiación en una posición a partir de la radiación en un punto que se toma como referencia. De modo que, a la hora de explotar el modelo, se debe elegir una estación de referencia entre todas las del conjunto  $E_R$ , que será la que suministre la radiación de entrada. Puesto que existen varias posibilidades, debemos determinar cuál de ellas será la más adecuada. Para tomar esta decisión, validamos nuestro modelo respecto a varios conjuntos de datos: uno por cada estación de  $E_R$ . De forma que, el conjunto de validación  $D_n$ , corresponde a tomar como referencia la estación  $e_n$ , de manera que estará formado por ejemplos cuyas componentes  $r_i$  son las radiaciones medidas en la estación  $e_n \in E_R$  y sus componentes  $r_o$  son obtenidas de cualquier otra estación  $e_m \in E_R$  con  $e_n \neq e_m$ . Por otro lado, puesto que, como se ha dicho anteriormente, no hemos reservado una parte de los ejemplos obtenidos a partir de la tabla SEMANA para la validación, hemos optado por utilizar los datos de la tabla DIA. Así que, en la tabla 2 mostramos el error para cada conjunto de validación. La primera columna muestra el código de la estación de referencia. En la segunda columna se muestra la cantidad de datos del conjunto. La tercera columna indica el RMSE obtenido al tratar de inferir la radiación en el resto de estaciones de  $E_R$  (es decir, dejando fuera la estación de referencia). Como vemos, el menor RMSE se alcanza cuando se utiliza la estación CR12 como referencia, mientras que utilizar AL62 conlleva el mayor error. Esto es lógico teniendo en cuenta la situación geográfica de estas dos estaciones (ver figura 4). AL62 está situada en la costa sur de la región mientras que CR12 está en una situación más céntrica, al igual que MO31, que también proporciona una de las medidas de error más bajas. En la figura 5 se muestran las medidas de radiación para un año inferidas por el modelo para AL62 utilizando como estación de referencia CR12 (línea de rombos) frente a las medidas reales (línea sólida).

**Interpolación de la Radiación Solar a partir de Varias Estaciones de Referencia.** Un enfoque diferente podría ser la utilización de varias estaciones

**Cuadro 2.** Errores de validación cuando se utiliza como radiación de entrada la proporcionada por cada una de las estaciones de  $E_R$

Código estación $e_n$	N° datos $D_n$	RMSE
AL62	6979	<b>28.71</b>
CI52	6389	25.41
CA91	6455	25.14
CR12	6023	<b>23.03</b>
JU12	6820	26.82
JU51	6860	25.63
JU61	7015	27.08
LO61	6365	25.65
MO31	6559	23.78
MO41	4880	24.48
MU62	6452	25.07
TP11	4360	27.46
TP42	6859	24.96



**Figura 5.** Radiación semanal real de la estación AL62 durante un año frente a la inferida utilizando CR12 como estación de referencia.

de referencia, de modo que, la radiación en cierto punto se obtendría agregando los valores de cada una de ellas.

Consideremos de nuevo el conjunto de todas las estaciones climáticas capaces de medir la radiación,  $E_R$  y sea  $c$  su cardinal. Como dijimos en la sección anterior, disponemos de las coordenadas de cada estación  $e_i \in E_R$  con  $i = 1, \dots, c$ , en el espacio  $DEM = X \times Y \times Z$ . Podemos asumir que los puntos  $e_i \in E_R$  son los centroides de  $c$  clusters difusos  $E_i$  con  $i = 1, \dots, c$ , en los que el espacio  $DEM$  está dividido. Si tomamos la definición del método FCM del grado de pertenencia de un punto  $p \in DEM$  al cluster  $E_i$ , se tiene que:

$$\mu_{E_i}(p) = \left( \sum_{j=1}^c \left( \frac{\|p - e_i\|}{\|p - e_j\|} \right)^{\frac{2}{m-1}} \right)^{-1} \quad (15)$$

con  $e_i, e_j \in E_R$ . Podemos asumir también, que cada cluster  $E_i$  define el antecedente de una regla TSK de orden 0,  $R_i$ , de la forma:

$$R_i : \text{If } p \text{ is } E_i \text{ then } r = rad_i$$

siendo  $r$  la variable de salida que representa la radiación y  $rad_i$  el valor *crisp* correspondiente a la radiación medida en la estación  $E_i$ . Teniendo en cuenta el factor temporal, cada regla será de la forma:

$$R_i(t) : \text{If } p \text{ is } E_i \text{ then } r = rad_i(t)$$

De modo que  $R_1(t), R_2(t), \dots, R_c(t)$  forman un modelo difuso TSK cuya salida para un punto  $p$  será:

$$r(p) = \frac{\sum_{i=1}^c \mu_{E_i}(p) rad_i(t)}{\sum_{i=1}^c \mu_{E_i}}$$

De manera que lo que se intenta es usar la información disponible (es decir, coordenadas de cada  $e_i$ , y los correspondientes valores de radiación) para formar un modelo de reglas difusas, y a partir de él y su mecanismo de inferencia, calcular la salida de radiación dado un punto  $p \in DEM$ .

### Adquisición de datos

Para validar un modelo obtenido siguiendo el método que acabamos de describir, vamos a seguir un enfoque *holdout* [14]. De esta manera, no debemos utilizar todo el conjunto  $E_R$  para generarlo, ya que algunas de ellas deben reservarse para la fase de validación. Por tanto, tenemos que dividir  $E_R$  en dos partes: una formada por las estaciones a partir de las cuales se van a definir las reglas difusas del modelo, que vamos a denotar  $I_R$  y otra, que vamos a llamar  $T_R$ , formada por las estaciones reservadas para la validación. Con el objeto de que el conjunto de estaciones para definir las reglas,  $I_R$ , esté formado por las

estaciones más representativas, decidimos aplicar el método SCM (ver 3.1) al conjunto  $E_r$ . Como sabemos, este método considera como candidatos todos los elementos del conjunto de datos de entrada. Así, si el conjunto de entrada es  $E_R$ , cada  $e_i \in E_R$  es un candidato a centroide. Finalmente, de ellos determina cuáles minimizan su función de error. Por tanto, los centroides que obtiene como salida siguen siendo puntos del  $DEM$  en los que existe una estación capaz de medir la radiación, y son dichas estaciones las que forman  $I_R$ . En la tabla 3 resumimos los valores empleados en el algoritmo SCM. La cardinalidad del conjunto  $E_R$

**Cuadro 3.** Parámetros de configuración para el método SCM en el experimento de la radiación.

Parámetro	Significado	Valor
$R_a$	Radio que define vecindad de influencia en potencial	0.25
$R_b$	Radio que define la vecindad grandes reducciones de potencial	0.375
$\epsilon_{min}$	Umbral de potencial bajo el cual se rechaza el punto como cluster	0.15
$\epsilon_{max}$	Umbral de potencial sobre el cual se acepta el punto como cluster	0.5

de las estaciones capaces de medir la radiación es  $c = 29$ . Aplicando SCM a  $E_R$  obtenemos un conjunto de clusters  $I_R$  con  $n = 13$  centroides. De modo que nuestro sistema de reglas difusas estará compuesto por 13 reglas.

Finalmente, para formar el conjunto de validación vamos a tener en cuenta las estaciones de  $T_R = E_R - I_R$ , siendo 16 su cardinal.

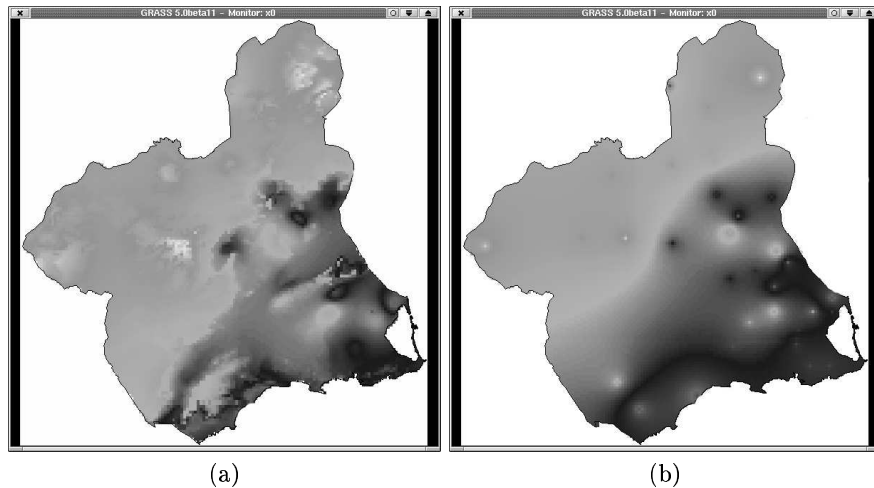
**Resultados de la Interpolación de la Radiación.** El RMSE medio alcanzado cuando se intentan inferir valores de radiación para las estaciones de  $T_R$  a partir del modelo generado en base a  $I_R$  es de 16.42. Siguiendo el enfoque de Chiu en su trabajo [6], podemos calcular el grado de pertenencia utilizando la definición derivada del método SCM:

$$\mu_{E_i}(p) = e^{-\alpha \|p - e_i\|^2}$$

siendo  $\alpha = \frac{4}{R_a^2}$ , teniendo que  $R_a$  es el radio fuera del cual, los puntos tienen poca influencia en  $e_i$  (ver sección 3.1). De este modo se obtiene un RMSE medio de 17.76.

Por otro lado, consideremos el trabajo [4], que resuelve el mismo problema disponiendo de la misma información procedente de las estaciones del conjunto  $I_R$ . Su enfoque se basa en la generación de la superficie de radiación mediante una red de Hopfield [11], corregida por medio de las salidas inferidas por un MLP. El funcionamiento de la red de Hopfield para generar la superficie de

radiación es el siguiente: se considera que el espacio  $X \times Y$  está dividido en una rejilla compuestas por celdas cuadradas. Entre todas las celdas, hay algunas (atractores) de las que se conoce su radiación, por contener una estación capaz de medirla. Por el contrario, existen celdas en las que ésta no se conoce (celdas libres) y debe calcularse a partir de la medida en los atractores. Para determinar la radiación en las celdas libres se lleva a cabo un proceso iterativo que, en el inicio, asigna a las celdas libres la media aritmética de la radiación medida en las estaciones de  $E_R$  (es decir, en los atractores). Luego, en cada iteración, se recalcula el valor de radiación de cada celda libre como la media aritmética de los valores de sus cuatro celdas vecinas. El proceso se repite hasta que se alcance un número máximo de iteraciones o hasta que el cambio medio de las celdas libres sea menor que cierto umbral. Debido a que no se tiene en cuenta la altura (es decir, se trabaja en el espacio  $X \times Y$  en lugar de en el espacio  $DEM$ ) los resultados de la red de Hopfield no son satisfactorios. Esto se ilustra en la figura 6 donde se muestran las superficies de radiación generadas por ambos métodos. La figura 6(a) muestra la radiación para cierto día debida al modelo difuso con la definición del grado de pertenencia dada por la fórmula 15, mientras que la figura 6(b) muestra la radiación generada por el modelo de Hopfield para el mismo día. Las zonas más oscuras corresponden a medidas de radiación más alta. La primera muestra unos cambios de radiación bruscos coherentes (según los expertos) con los accidentes del terreno. Mientras que en la segunda superficie, las transiciones siempre son suaves haya o no accidentes geográficos en la zona.



**Figura 6.** Superficies de radiación generadas para un día determinado: (a) mediante el método difuso; (b) mediante el método híbrido.

Para solucionar el problema del enfoque de Hopfield, se opta por corregir la superficie de radiación sumando en cada punto las salidas inferidas por un MLP previamente entrenado para ello. Sin embargo, dicho MLP requiere un sofisticado proceso de adquisición de los datos que consiste en seleccionar los valores de radiación en las estaciones de  $I_R$  para un conjunto de días, generar  $c$  superficies de Hopfield para cada uno de esos días, y calcular la diferencia entre el valor otorgado por la red de Hopfield y la radiación real en cada celda y para cada día. Para más información ver el trabajo [4]. Una vez entrenado el MLP se obtiene un RMSE de 16.40. Como vemos supera al mejor de nuestros dos modelos difusos solamente en dos centésimas. Esta pequeña diferencia no compensa el costoso proceso de adquisición de los datos requeridos para entrenar la MLP, frente a la obtención directa de nuestro modelo difuso de radiación que no requiere ningún proceso de DDFM si no que se deriva simplemente de la conversión a difuso de las coordenadas  $DEM$  de las estaciones capaces de medir la radiación. Por otro lado, a la hora de explotar el modelo, la inferencia de la radiación en un punto por parte del modelo difuso es directa. En el otro enfoque sin embargo, hay que generar una superficie de Hopfield por cada estación con radiación antes de poder inferir una salida.

En la tabla 4, se muestran los errores para cada estación de  $T_R$ , de los dos modelos difusos y del enfoque Hopfield + MLP. Podemos observar que el mayor error se obtiene para JU12. Esto se debe al hecho de que JU12 casi siempre registra medidas de radiación por encima del resto de estaciones, lo que probablemente se debe a algún fallo en los sensores de radiación.

Por otra parte, recordemos que se dispone del conjunto  $E_R$  de estaciones capaces de medir la radiación. En el caso del enfoque difuso, en lugar de utilizar todas las estaciones de  $E_R$  para definir el modelo, solamente se usa el conjunto  $I_R \subset E_R$  y se reserva  $T_R \subset E_R$  para validarlo. Por tanto, estos resultados son pesimistas puesto que la utilización de todo  $E_R$  para formar el modelo para su posterior uso en un entorno real permitiría más capacidad aproximativa al utilizar información de radiación procedente de más puntos de la Región de Murcia. Del mismo modo, en el enfoque Hopfield + MLP, solamente se emplean como atractores y más tarde, en el proceso de entrenamiento del MLP, las estaciones del conjunto  $I_R$  con lo que se está ignorando la información procedente de las estaciones de  $T_R$  para generar el modelo. Con esto queremos decir que la utilización de datos procedentes de todas las estaciones de  $E_R$  dará lugar a un modelo con mayor capacidad aproximativa.

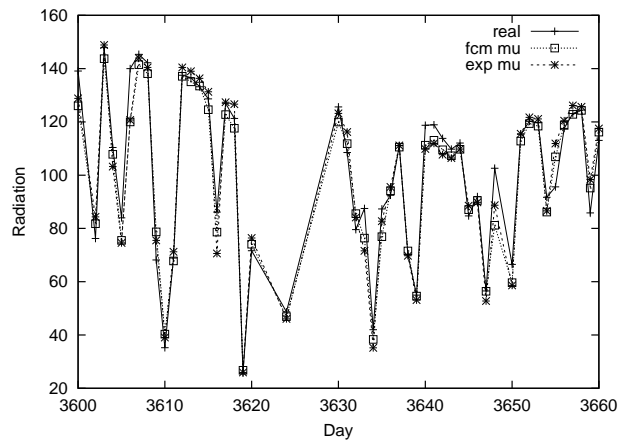
Finalmente, en la figura 7, se muestra la radiación inferida en la estación MO61 utilizando el método difuso con las distintas definiciones de  $\mu$  y la radiación real. Podemos ver que en este caso, casi siempre la radiación inferida está por debajo de la real, pero siempre muy próxima a ella.

#### 4.2. Fusión de Sensores Sonar e Infrarrojos

En esta sección vamos a ver un ejemplo de aplicación del DDFM para la navegación de robots móviles. El objetivo final es que un robot móvil sea capaz de navegar en entornos interiores sin conocimiento previo sobre la estructura de

**Cuadro 4.** Valores de RMSE para cada estación de  $T_R$

Código estación	N° datos	RMSE		
		$\mu$ FCM	$\mu$ EXP	Hopfield + MLP
AL31	366	12.30	14.70	13.21
AL51	366	15.42	17.43	15.41
CA21	366	13.24	12.85	12.74
CA52	353	14.22	12.06	12.72
CA72	366	16.78	15.11	17.44
CA91	363	15.33	15.02	15.53
CI32	366	15.18	20.56	14.25
JU12	364	32.78	34.54	36.10
LO61	365	18.05	25.07	12.45
MO12	360	16.93	17.52	17.32
MO41	360	11.03	11.48	11.70
MO51	366	15.00	13.68	13.82
MO61	366	10.88	12.84	10.82
MU31	364	16.79	17.94	15.91
MU62	271	13.23	13.89	13.17
TP81	279	11.55	11.64	11.36



**Figura 7.** Radiación real y radiación inferida para la estación MO61 cuando se utilizan  $\mu$  FCM y  $\mu$  EXP, para un periodo de dos meses.



la planta y sin modificación del entorno. Una de las tareas a resolver para poder alcanzar ese objetivo es la llamada *fusión de sensores*. Los robots móviles recogen información a través de sus *sensores externos* (sónar, infrarrojos, sensores láser, cámaras de visión, etc.). Esta información junto con la proporcionada por sus sensores *internos* (por ejemplo, sensores holométricos), es utilizada por el robot para construir su propio mapa del entorno, de modo que, cuanto más fiable sea esta información, mejor será el mapa. En concreto, nuestro robot dispone de sensores sónar e infrarrojos de bajo coste, cuyas salidas son medidas de distancias. Sin embargo, las lecturas que proporciona cada tipo no son lo suficientemente fiables debido a diversas fuentes de error. No obstante, combinar información inexacta procedente de varios sensores produce una reducción del error de la información global. Se llama *fusión de sensores* al proceso mediante el cual se combinan datos de diferentes sensores para obtener lo que podríamos considerar las medidas dadas por un *sensor virtual*.

Existen diversos métodos para fusión, como los filtros de Kalman extendidos [15], los métodos basados en reglas [9], e incluso las redes neuronales artificiales (ANN) [5,2]. En concreto, nosotros vamos a resolver el problema por medio de un modelo de reglas difusas obtenido mediante un proceso de DDFM. De modo que, suponiendo que existe una función:

$$f_{FUSION} : SONAR \times IR \rightarrow DISTANCIA_{REAL}$$

que a partir de la lectura del sónar y la lectura del sensor infrarrojo (*IR*) obtuviera la distancia real, nuestro problema se traduce a generar un modelo de reglas difusas que aproxime  $f_{FUSION}$ .

**Adquisición de Datos y Fuentes de Error.** Uno de los problemas de la aplicación del DDFM para la fusión de sensores, es la adquisición de los datos. Ésta no es una tarea simple, ya que se necesitan, tanto lecturas de los sensores como medidas reales suficientes para diferentes ángulos de incidencia, superficies de objetos y distancias. Además nuestro robot está dotado de varias parejas de sensores sónar e infrarrojo. Los sensores de una pareja tienen la misma orientación en el robot, de modo que se obtiene información redundante procedente de dos fuentes distintas. Todo esto implica que la preparación de los experimentos adecuados en un entorno real sea muy difícil y lleve mucho tiempo. Por tanto, y puesto que se encuentran disponibles en la literatura especializada buenos modelos de los sensores, se ha optado por la simulación a la hora de la adquisición de los datos. Sin embargo, para que esta simulación sea lo más parecida a la realidad posible hay que tener en cuenta las fuentes de error que afectan a estos sensores.

El tipo de sónar que tiene nuestro robot es de la serie *Polaroid 6500 series Ultrasonic Sonar RangeFinder*. Éste envía y recibe pulsos ultrasónicos con una frecuencia de 49.4 kilohercios. Tales pulsos tienen su máxima intensidad en un ángulo de entre 15 y 20 grados. Puede medir distancias entre 15 centímetros y 10 metros con una precisión de  $\pm 1\%$  en todo el rango. Destacamos las siguientes fuentes de error:

- Pérdida de la señal de eco: cuando el ángulo con el que incide la señal en el objeto es suficientemente grande, los pulsos no regresan al receptor y por tanto el sensor no detecta el objeto.
- Incertidumbre asociada al rango angular: la señal ultrasónica emitida forma un cono con vértice en el s3nar. Cuanto m3s alejado est3 el objeto en el que incide, m3s incertidumbre se asocia a la medida devuelta por el s3nar ya que, no se puede saber exactamente cu3l es su posici3n debido al ancho de ese cono, que es mayor cuanto m3s alejado est3 el objeto.
- Dobles rebotes: a veces, cuando el 3ngulo de incidencia de la se3al sobre un objeto es lo suficientemente grande, el pulso rebotado, puede alcanzar otro objeto y volver al sensor. En este caso, se detecta el objeto pero se produce una medida mayor de la que es en realidad (ya que, al rebotar el pulso hacia otro objeto, tarda en volver m3s tiempo).

Para simular estas fuentes de error se implementa una t3cnica basada en el m3todo de trazado de rayos de infografía.

En cuanto a los sensores infrarrojos, son de la serie *Sharp GPD2D02 High Sensitive Distance Measuring Sensor*. La anchura de m3xima intensidad del rayo es de 10 cent3metros. Puede medir distancias entre 10 y 120 cent3metros con una precisi3n y resoluci3n dependiente de la distancia. Las principales fuentes de error son:

- Color/reflectividad del objeto: como la reflectividad de los objetos var3a con el color, cuanto m3s oscuro es un objeto, m3s distante parece. La precisi3n se degrada tanto con la distancia de un objeto oscuro, que puede incluso llegar a no detectarse.
- Iluminaci3n global: puesto que la iluminaci3n ordinaria contiene luz en la banda infrarroja, cuanto m3s brillante es la iluminaci3n, m3s lejano parece el objeto.
- Error de cuantizaci3n: este sensor convierte su lectura a 8 bits. Esta conversi3n no es lineal en todo el rango y por tanto, cuanto m3s lejano est3 el objeto, m3s imprecisa es la lectura.

Para simular las fuentes de error del sensor infrarrojo, se ha construido directamente el modelo proporcionado por el fabricante.

Se han usado dos escenarios tanto para recoger los datos de entrenamiento como de validaci3n. El primer escenario es la planta s3tano de nuestra facultad. Tiene 17 metros de ancho por 22 de largo, con pasillos de 2 metros de ancho. Los umbrales de las puertas miden 80 cent3metros. El segundo escenario es una sala del *Trinity College* con una dimensi3n de 3.5×3.5 metros, y pasillos y puertas de 65 cent3metros de ancho.

La adquisici3n de los datos consiste en la obtenci3n de triplas de medidas { *s3nar, infrarrojo, real* } en diferentes posiciones y orientaciones a lo largo de dos caminos predefinidos, obteniendose el conjunto de datos que, aunque se haya generado mediante simulaci3n, posee muchos valores contradictorios como ocurre en el mundo real.

**DDFM para la Fusión de Sensores.** Vamos a generar el modelo difuso mediante una estrategia híbrida de DDFM. Para ello, utilizamos el 70% de los datos para entrenamiento, reservando el 30% para validación. En primer lugar, se detecta el número de reglas que va a tener el modelo mediante el algoritmo de clustering auto-organizativo MCM'. Puesto que este algoritmo, finalmente, selecciona los centroides de entre los datos de entrenamiento, es conveniente aplicar el algoritmo de clustering HCM (ver sección 3.2) para optimizarlos. De modo que HCM se inicializa con los centroides detectados mediante el método MCM' y los va ajustando de manera que se minimice su función de costo. Finalmente, los centroides detectados en la fase anterior son los parámetros iniciales de una red ANFIS que genera un modelo de reglas de tipo TSK. En la tabla 5 se muestran los parámetros empleados en la configuración de cada una de estas técnicas.

**Cuadro 5.** Parámetros para la estrategia híbrida empleada en el experimento de fusión de sensores

Parámetro	Significado	Valor
MCM' $\sigma$	altura y suavidad de la función montaña	1
MCM' $\beta$	eliminación del efecto del último cluster	1.5
MCM' $\delta$	umbral de tolerancia	0.001
FCM $m$	<i>fuzziness</i>	2
FCM $\epsilon$	umbral de tolerancia	0.001
ANFIS épocas	iteraciones del proceso de aprendizaje	10000
ANFIS $a$	parámetro $a$ de la MF <i>bell-shaped</i>	0.5
ANFIS $b$	parámetro $b$ de la MF <i>bell-shaped</i>	2

Una vez que se ha llevado a cabo el proceso de entrenamiento, vamos a medir la capacidad de generalización del modelo difuso. De modo que se sitúa de nuevo al robot en ambos escenarios recorriendo otros caminos predefinidos y tomando medidas de sus sensores. El RMSE entre las medidas fusionadas mediante las reglas difusas y las medidas reales se muestra en la tabla 6. También mostramos los errores obtenidos para la fusión mediante reglas difusas con los errores para las distancias medidas por el sensor infrarrojo (IR) y los debidos a las lecturas del sónar. Además vamos a compararlos con otros mecanismos de fusión. En primer lugar con uno ampliamente utilizado como es el modelo de Flynn [9]. Las reglas que forman este modelo son las siguientes:

- Si la lectura del sónar es mayor que el rango máximo del sensor infrarrojo, entonces ignorar la medida de éste último.
- Si la lectura del sónar está en su máximo valor, entonces la distancia real es mayor.
- Cuando el sensor infrarrojo presenta un cambio de no detección a detección y la lectura del sónar es menor de 3 metros entonces se ha detectado un objeto.

Este modelo funciona bien en simuladores pero no en el mundo real debido a las dos siguientes razones. En primer lugar, no tiene en cuenta que ha mejorado sensiblemente la precisión de los sensores infrarrojos modernos. Por tanto, los métodos de fusión deben dar más importancia al sensor infrarrojo para obtener más precisión en las distancias cortas. Por otro lado, el método de Flynn no tiene en cuenta las características físicas de cada par de sensores. El proceso de fabricación no es uniforme y se producen diferencias de calibración entre los sensores que deberían tenerse en cuenta.

Finalmente también mostramos el RMSE debido al mecanismo de fusión obtenido al entrenar un MLP con 10 nodos ocultos durante 20000 épocas. Podemos observar que los mejores errores se obtienen para el modelo difuso TSK para el escenario *Trinity*. Para el escenario *Facultad*, se obtiene resultados idénticos para el MLP y para el modelo TSK.

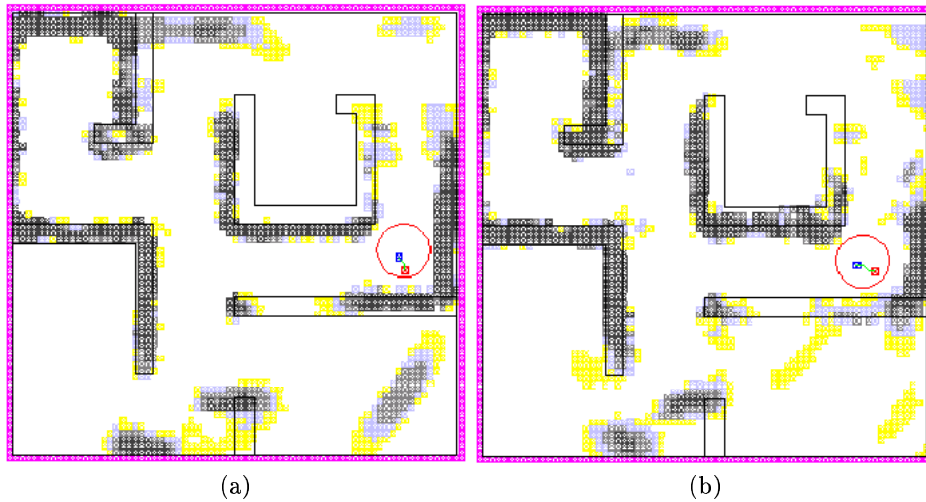
**Cuadro 6.** Medidas de error de validación

Método de fusión	RMSE Trinity	RMSE Facultad
SONAR	4.01	5.08
IR	0.51	5.05
Flynn	0.67	3.77
MLP	0.66	3.38
MCM'+HCM+ANFIS	0.47	3.38

El último paso, es validar el modelo difuso de fusión en la fase de construcción del mapa del entorno. El mapa está dividido en celdas cuadradas, cada una de las cuales tiene asociado un grado de certeza de que la celda esté vacía y un grado de certeza de que esté ocupada. En la figura 8(a) podemos ver el entorno *Trinity* en comparación con el mapa generado por el robot partiendo de las distancias debidas a las reglas de Flynn. Mientras que en la figura 8(b) podemos ver el mapa generado a partir de las reglas difusas. Cuanto más oscura aparece una celda, mayor es el grado de certeza de que esté ocupada. Podemos ver por ejemplo que, partiendo de las reglas de Flynn, se detecta un objeto en la esquina inferior derecha que no existe. Sin embargo, en el mapa generado a partir de las reglas difusas, ese objeto inexistente no aparece.

## 5. Conclusiones

El modelado difuso es un enfoque relativamente novedoso para la construcción de modelos de sistemas utilizando un lenguaje basado en la lógica difusa con predicados difusos. Cuando existen datos entrada-salida se pueden utilizar técnicas de identificación de sistemas para realizar el proceso de modelado: es lo que se denomina Modelado Difuso Conducido por Datos (DDFM). La naturaleza modular de este proceso favorece la aplicación de combinaciones de técnicas cada una de ellas resolviendo una de sus fases. Concretamente, en este capítulo hemos



**Figura 8.** Mapas del entorno *Trinity*: (a) generado a partir de la fusión de sensores usando las reglas de Flynn; (b) generado a partir de la fusión de sensores mediante reglas difusas.

abordado el proceso mediante estrategias compuestas por técnicas de clustering auto-organizativo para la elección del número de reglas, técnicas como HCM y FCM para la generación de éstas y la red neuro-difusa ANFIS para el ajuste de sus parámetros.

Para probar la validez de este enfoque, hemos expuesto tres ejemplos de aplicación procedentes de diferentes contextos. En primer lugar, hemos abordado el problema de la estimación de la evapotranspiración de referencia, magnitud fundamental a la hora de determinar las necesidades hídricas de un cultivo. Con los enfoques difusos hemos obtenido unos resultados que superan con creces a los del modelo de evapotranspiración más utilizado tradicionalmente e incluso a los de un perceptrón multicapa.

El siguiente problema tratado ha sido el de la interpolación de la radiación solar. Lo abordamos siguiendo dos alternativas de modelado difuso: una generando el modelo a partir de los datos y otra directa, en la que las propias estaciones (o sus coordenadas) se consideran los antecedentes de las reglas y sus respectivos valores de radiación los consecuentes. Esta última alternativa resulta mejor que la primera al utilizar toda la información de radiación disponible y no sólo la procedente de una única estación.

Por último se ha utilizado la técnica híbrida de DDFM para la obtención de un modelo de fusión de sensores, cuyos resultados mejoran a otro modelo ampliamente usado como las reglas de Flynn y que igualan los de un perceptrón multicapa.

Las experiencias expuestas, ponen de manifiesto que, el DDFM en general y las técnicas de DDFM híbridas en particular, resultan un enfoque válido y adecuado para el modelado de sistemas complejos a partir de datos entrada-salida.

## Referencias

1. R. Babuška. *Fuzzy Modeling and Identification*. PhD thesis, Delft University of Technology, The Netherlands, 1997.
2. H. Martínez Barberá, A. Gómez Skarmeta, M. Zamora Izquierdo, and J. Botía Blaya. Neural networks for sonar and infrared sensors fusion. In *Fusion 2000, Paris, July 2000*, 2000.
3. J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum, New York, USA, 1981.
4. J. A. Botía Blaya, A. M. Padilla Urrea, A. F. Gómez Skarmeta, and M. Valdés Vela. Soft computing aplicado a la interpolación gis. In *Actas de la IX Conferencia de la Asociación Española de Inteligencia Artificial*, Gijón, Noviembre 2001. CAEPIA.
5. C. C. Chang and K. T. Song. Ultrasonic sensor data integration and its application to environment perception. *Journal of Robotic Systems*, 13(10):663–677, 1996.
6. S.L. Chiu. Fuzzy model identification based on cluster estimation. *Journal of the Intelligent and Fuzzy Systems*, 2(3):267–278, 1994.
7. K. B. Cho and B. H. Wang. Radial basis function based adaptive fuzzy systems and their application to system identification and prediction. *Fuzzy Sets and Systems*, 83:325–339, 1996.
8. H. Choe and B. Jordan. On the optimal choice of parameters in a fuzzy c-means algorithm. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, pages 349–354, 1992.
9. A. M. Flynn. Combining sonar and infrared sensors for mobile robot navigation. *Intl. Journal Robot. Res.*, pages 5–14, 1988.
10. K. Hirota and W. Pedrycz. Fuzzy computing for data mining. In *Proceedings of the IEEE*, volume 87, pages 1575–1601, September 1999.
11. J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of the National Academy of Science*, pages 3088–3092, USA, 1982.
12. J.-S. R. Jang, C.-T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing*. Matlab Curriculum. Prentice Hall, 1997.
13. J.S. Jang. Anfis: Adaptive-network-based fuzzy inference systems. *IEEE Trans. on Systems, Man, and Cybernetics*, 23(03):665–685, 1993.
14. Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, pages 1137–1145, 1995.
15. J. Leonard and H. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer, Dordrecht, The Netherlands, 1992.
16. Smith M., Allen R. G., and Pereira L. Revised fao methodology for crop water requeriment. In *Proceedings of the International Conference on Evapotranspiration and Irrigation Scheduling*, San Antonio., Texas, 1996.
17. J. L. McClelland and D. E. Rumelhart. *Explorations in Parallel Distributed Processing*. MIT Press, 1988.
18. N. R. Pal, J.C. Bezdek, and E.C.K. Tsao. Generalized clustering networks and kohonen's self-organizing schemes. *IEEE Trans. On Neural Networks*, 4(4):549–557, July 1993.

19. A. F. Gómez Skarmeta. *Modelado Difuso de Sistemas Mediante Aprendizaje por Clasificación de Técnicas de Agrupamiento (Clustering)*. PhD thesis, Universidad de Murcia, 1995.
20. M. Sugeno and G. T. Kang. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28(1):15–33, 1988.
21. T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetic*, 15:116–132, 1985.
22. P. Vuorimaa. Fuzzy self-organizing map. *Fuzzy Sets and Systems*, 66:223–231, 1994.
23. R. Yager and D. Filev. Approximate clustering via the mountain method. *IEEE Transactions on Systems, Man, and Cybernetics*, 24:1279–1284, 1994.