On-line Learning With Reject Option

G. J. Pérez, M. Santibáñez, R. M. Valdovinos, J. R. Marcial, M. Romero, R. Alejo

***Abstract*— On-line learning is a training paradigm that allows the processing of constant data flows, so that learning adapts to new knowledge. However, due to the nature of the study problem, it is possible that in the clustering obtained there are data complexities (outliers, atypical patterns, noisy, etc.) that deteriorate the performance of the model in the classification stage. Due to the above, an alternative to cope data complexities is the use of algorithms that allow to detect reject options to filter noisy pattern. In this research the neighborhood-based reject option is implemented in an on-line learning process, with the intention of improving the clustering quality and thus increasing the precision indexes obtained with the nearest neighbor's rule in the classification stage. Likewise, to validate the quality of the clustering generated, internal and external analysis metrics are used. The experimental results show the viability of the proposal when analyzed on real data.**

***Keywords*— Preprocessing, On-line Learning, Clustering, Classification, Data Mining.**

I. INTRODUCCIÓN

EL APRENDIZAJE *On-line* trabaja basado en flujo constante de datos, el cual debe ser procesado y clasificado en tiempo real [1], aprovechando el conocimiento generado mientras el clasificador realiza su trabajo. Para lograr lo anterior, el aprendizaje *on-line* se ejecuta sobre lotes de datos, en los que se presenta parte del total de los datos de entrenamiento, con los que el modelo realiza la clasificación de datos nuevos [2].

En su funcionamiento general, el aprendizaje *on-line* considera que no se tiene conocimiento sobre la distribución en clases de sus datos de entrada, por lo que con la ayuda de algoritmos de agrupamiento o *clustering* se logra identificar de forma automática la distribución de los datos en grupos, mismos que se transformarán en clases. Posteriormente, conforme van llegando más patrones, éstos son incorporados al grupo con el que se identifique mayor semejanza. Finalmente, cuando se tiene un patrón nuevo que se desea clasificar, se hace uso de la distribución obtenida hasta el momento para realizar el reconocimiento hacia el grupo con que muestre mayor semejanza [3].

Para validar la calidad del agrupamiento obtenido, en la literatura se proponen métodos de análisis de condensado y separabilidad de grupos. No obstante, dependiendo de la distribución de los datos, se pueden obtener agrupamientos con complejidades de datos (desbalance, solapamiento, *outliers*, etc.) que pueden deteriorar el rendimiento del clasificador en la etapa de reconocimiento [4].

Los algoritmos de preprocesado orientados a identificar, eliminar y/o disminuir los efectos negativos que las complejidades de datos tienen en el rendimiento del clasificador son los de limpieza [5], condensado [6] y reetiquetado de patrones también conocido como Opción de Rechazo [7].

Esta investigación se enfoca a analizar la pertinencia de utilizar algoritmos de limpieza del conjunto de datos (CD) posterior a un proceso de agrupamiento dentro de una estrategia de aprendizaje *On-line*. De forma específica, se aplicará un algoritmo de reetiquetado o rechazo basado en la regla de los K-vecinos, los cuales realizan el reetiquetado de aquellos patrones que se encuentren mal ubicados o bien el rechazo (eliminación) de patrones que presentan inconsistencia respecto a los grupos obtenidos por el algoritmo de agrupamiento.

II. MÉTODOS Y HERRAMIENTAS

2.1 Algoritmo Batchelor & Wilkins

Batchelor & Wilkins es un algoritmo heurístico incremental que emplea un único parámetro (θ), el cual representa la fracción de la distancia media entre agrupaciones, utilizado para calcular un umbral de distancia que determina si se crea o no un nuevo agrupamiento [8] tal como se observa en el Algoritmo 1.

Algoritmo 1: Batchelor & Wilkins
Entradas: θ = Fracción de la distancia media entre agrupaciones
1. Seleccionar un patrón al azar y convertirlo en el centro del primer agrupamiento. 2. Seleccionar el patrón más alejado del primer agrupamiento y convertirlo en el centro del segundo agrupamiento. 3. Calcular las distancias de todos los patrones a los centros y determinar la distancia mínima a un centro para cada patrón. 4. Obtener el patrón más alejado de los agrupamientos existentes (Máximo de las distancias mínimas de los patrones a los agrupamientos) 5. Si la distancia del patrón escogido es mayor que el umbral especificado, crear un nuevo agrupamiento con el patrón seleccionado. 6. Repetir los pasos 3, 4 y 5 hasta que ya no se creen nuevos agrupamientos. 7. Asignar cada patrón a su agrupamiento más cercano.

El algoritmo funciona de la siguiente manera: El primer centro se crea tomando un patrón al azar del CD, posteriormente un segundo centro es generado utilizando el patrón más alejado del primer centro. A partir de este punto, se obtiene el patrón más alejado a los grupos existentes, si la distancia del patrón escogido al conjunto de centros es mayor al umbral establecido (θ) se crea un nuevo grupo con dicho patrón, lo anterior se repite mientras se generen nuevos grupos. Finalmente se asigna cada patrón a su agrupamiento

G. J. Pérez, Universidad Autónoma del Estado de México, Facultad de Ingeniería. Cd. Universitaria, Toluca, México, gerardo-jpa@hotmail.com.

M. Santibáñez, Universidad Autónoma del Estado de México, Facultad de Ingeniería. Cd. Universitaria, Toluca, México, monicass_isc@hotmail.com.

R. M. Valdovinos, Universidad Autónoma del Estado de México, Facultad de Ingeniería. Cd. Universitaria, Toluca, México, li_rmvr@hotmail.com.

J.R Marcial, Universidad Autónoma del Estado de México, Facultad de Ingeniería. Cd. Universitaria, Toluca, México, jrmarcialr@uaemex.mx.

M. Romero, Universidad Autónoma del Estado de México, Facultad de Ingeniería. Cd. Universitaria, Toluca, México, mrh1601@yahoo.com.

R. Alejo, Instituto Tecnológico de Estudios Superiores de Jocotitlán, Jocotitlán, México, ralejoll@hotmail.com

Corresponding author: Gerardo Javier Pérez Álvarez

más cercano.

2.2 Índices de validación

Para analizar la calidad del agrupamiento la evaluación se puede realizar desde 3 enfoques [9]: Criterio Externo, Criterio Interno y Criterio Relativo.

2.2.1 Índices de Criterio Externo

Estos índices evalúan el agrupamiento resultante con base en información conocida a priori. Por ejemplo el número de grupos que se sabe se deben obtener con el algoritmo de *clustering*. Para ello, la validación parte de la comparación entre dos particiones de *C' y K' clusters*, $A = \{C_1, C_2, \ldots C_{C'}\}$ y $B = \{K_1, K_2, \ldots K_{K'}\}$.

(i) Recuento de pares. Las medidas a calcular están basadas en los resultados de la matriz de contingencia entre pares de agrupamientos [10]. Para esto, hace uso de los siguientes criterios:

C' = Número de clusters en la partición A

K' = Número de clusters en la partición B

N = Total de pares

n_{ij} = Número de pares en común entre la partición A y B

a. Los pares están en el mismo *cluster* en ambas particiones A y B (SS)

$$SS = \left(\frac{1}{2}\right)\sum_{i=1}^{K'}\sum_{j=i}^{C'} n_{ij}^2 - \left(\frac{N}{2}\right) \quad (1)$$

b. Los pares están en el mismo *cluster* en la partición A, pero en diferente *cluster* en B *(SD)*

$$SD = \left(\frac{1}{2}\right)\sum_{j=1}^{C'} n_{.j}^2 - \left(\frac{1}{2}\right)\sum_{i=1}^{K'}\sum_{j=i}^{C'} n_{ij}^2 \quad (2)$$

c. Los pares están en el mismo *cluster* en la partición B, pero en diferente *cluster* en A *(DS)*

$$DS = \left(\frac{1}{2}\right)\sum_{i=1}^{K'} n_{i.}^2 - \left(\frac{1}{2}\right)\sum_{i=1}^{K'}\sum_{j=i}^{C'} n_{ij}^2 \quad (3)$$

d. Los pares están en diferente *cluster* en ambas particiones K y C (DD)

$$DD = \frac{N(N+1)}{2} - \left(\frac{1}{2}\right)\left[\sum_{i=1}^{K'} n_{i.}^2 + \sum_{j=1}^{C'} n_{.j}^2\right] \quad (4)$$

(ii) Índice Rand [11], sirve para comparar la similaridad entre dos agrupamientos (A, B) mediante la medición del número de pares que estén en el mismo *cluster* (SS) o bien diferente *cluster* (DD). Este índice toma el valor entre 0 y 1. El valor 1 indica que ambas particiones (A y B) son idénticas.

$$Rand = \frac{SS+DD}{SS+SD+DS+DD} \quad (5)$$

(iii) Índice Jaccard [12], a diferencia del índice Rand no considera los pares que están en diferente *cluster* en ambas particiones (DD), puede tomar el valor entre 0 y 1.

$$Jaccard = \frac{SS}{SS+SD+DS} \quad (6)$$

(iv) Fowlkes & Mallows [13], al igual que el índice Jaccard no considera los pares que están en diferente *cluster* en ambas particiones (DD).

$$Fowlkes\ \&\ Mallows = \frac{SS}{\sqrt{(SS+SD)(SS+DS)}} \quad (7)$$

Tanto para el índice Rand, Jaccard y Fowlkes & Mallows entre más alto sea el valor obtenido la similaridad entre los dos agrupamientos es mayor.

(v) Índice Rand Ajustado [14], es una versión ajustada del índice Rand. Una problemática presente para el índice Rand es que el resultado del índice para dos particiones aleatorias no toma un valor constante e incrementa conforme el número de grupos. El ajuste que se realiza en este índice se asegura de que el valor sea cercano a 0 cuando las particiones no son similares o bien exactamente 1 cuando los agrupamientos son idénticos, esto sin depender del número de grupos.

$$ARI = \frac{\binom{n}{2}(SS+DD) - [(SS+SD)(SS+DS)+(SD+DD)(DS+DD)]}{\binom{n}{2}^2 - [(SS+SD)(SS+DS)+(SD+DD)(DS+DD)]} \quad (8)$$

(vi) La Medida F [15], interpretada como la media armónica entre los coeficientes de precisión y exhaustividad (*recall*) asume un valor en el intervalo entre 0 y 1, tiende a resultar en un valor alto cuando tanto la precisión como el *recall* son elevados. Esta medida permite determinar qué tan parecido es el agrupamiento resultante del algoritmo al agrupamiento real.

$$F\ Measure = \frac{2}{\left(\frac{1}{Precision}\right)\left(\frac{1}{Recall}\right)} \quad (9)$$

2.2.2 Índices de Criterio Interno

Estos índices evalúan la calidad del agrupamiento con base en la información presente en el conjunto de datos, pueden ser empleados para escoger el mejor algoritmo de agrupamiento así como el número de grupos óptimos sin utilizar información a priori.

(i) Índice Davies-Boulin, estima la similaridad de un grupo midiendo las distancias entre los patrones pertenecientes a un grupo respecto a su centroide y la disimilaridad entre los grupos con base en la distancia entre centroides [16]. El agrupamiento que genere el valor más bajo para este índice es considerado como la mejor solución.

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{j=1\ldots k; i<>k} \frac{d_i+d_j}{d(C_i,C_j)} \quad (10)$$

(ii) Índice Calinski-Harabasz, en este índice la similaridad de los grupos se define con base en la distancia de los patrones del grupo respecto a su centroide [17]. En tanto que la disimilaridad se define en base a la distancia de los centroides respecto al centroide global de todos los grupos generados [16]. Cuando se emplea este índice se busca el agrupamiento que genere el pico o valor máximo, de lo contrario, si forman una línea recta, ya sea ascendente o descendente, o una curva suave cualquier valor puede ser considerado como aceptable.

$$CH = \frac{trace(S_B)}{trace(S_W)}\left(\frac{n_p-1}{n_p-k}\right) \quad (11)$$

2.3 Opción de Rechazo

Los algoritmos de Opción de Rechazo, analizan si un determinado patrón se ha ubicado de forma correcta en el grupo o clase donde se encuentra, si debería de moverse de grupo o definitivamente ser rechazado (eliminado).

Este algoritmo tiene una ventaja respecto a otros algoritmos, permite al usuario ajustar la relación de error-rechazo, mediante el establecimiento de un umbral, para que se adapte de acuerdo al problema que se enfrenta.

En el contexto de aprendizaje *On-line*, la Opción de Rechazo tiene por objeto servir como filtro para decidir qué patrones deben ser incorporados y cuáles no, es decir, desempeñan la función de evaluador de patrones, en donde los patrones que estén lo suficientemente cerca del espacio de representación de la clase/grupo (definido por el umbral) a la cual se ha decidido incorporar serán aceptados.

En esta investigación se utiliza el algoritmo de Opción de Rechazo "Regla 2NN Modificada" [7], el cual tiene por

objetivo identificar si la asignación de un determinado patrón a los grupos formados por el algoritmo de agrupamiento es correcta, dependiendo del grupo al que pertenecen sus vecinos más cercanos. (Ver Algoritmo 2).

Algoritmo 2: Regla 2NN Modificada
Entradas: M = Conjunto de Patrones, $\{x_i \mid i=1,2,\ldots m\}$ K = 2 vecinos a considerar T = [0.8, 0.95]
Salidas: CEM = Conjunto de Datos Modificado
Algoritmo: Para todo i = 2 hasta i = m hacer Buscar los k vecinos de x_i en M-$\{x_i\}$ Si las etiquetas de $((k_1 = k_2) <> x_i)$ entonces Cambiar la etiqueta de x_i por la de los vecinos Caso contrario $h = d(x_i, k_1)/d(x_i, k_2)$ Si $h < T$ entonces x_i se asigna a la clase de k_1 Caso contrario x_i se rechaza Fin Si Fin Si Fin Para todo

El valor del umbral (T) puede oscilar entre el rango de 0.8 a 0.95, se decide si el patrón es recolocado o desechado dependiendo de si la fracción de la distancia entre el patrón y los dos vecinos más cercanos (h) es menor o mayor al umbral establecido, respectivamente.

III. METODOLOGÍA

La metodología propuesta para implementar la opción de rechazo es un proceso de aprendizaje *On-line* [18], se define en las siguientes etapas:

(i) Primera etapa. Obtención del agrupamiento del CD no etiquetado, con el algoritmo Batchelor & Wilkins, variando los valores del parámetro θ en: 0.1, 0.2, 0.3, 0.5, 0.7 y 0.9.

(i) Obtención de los índices de validación de Criterio Interno, Davies-Boulin y Calinski-Harabasz. Dado que no se conoce el número de grupos a priori para los conjuntos de datos Joensuu y MOPI, únicamente se realizó el análisis de los Índices de Validación de Criterio Externo basados en el recuento de pares para el CD D31.

(ii) Segunda etapa. Realizar validación cruzada utilizando el 80% de los patrones para fines de entrenamiento y el 20% restante para prueba. Posteriormente utilizar la Regla K-NN para clasificación con K= 3.

(iii) Tercer Etapa. Ejecución de la Regla 2NN Modificada en dos etapas diferentes: posterior a la ejecución del algoritmo de *clustering* (primera etapa) y posterior a la ejecución de la Validación Cruzada (segunda etapa).

IV. RESULTADOS

Los resultados mostrados en esta sección se agrupan en dos partes: resultados experimentales implementando el algoritmo de rechazo posterior al agrupamiento de datos y los resultados ejecutando la opción de rechazo posterior a la validación cruzada. En adelante se empleara la siguiente notación:

- SOR – Sin Opción de Rechazo
- COR – Con Opción de Rechazo

Con la finalidad de validar los resultados obtenidos, se tomaron como valor de referencia los resultados obtenidos en [18] y [19], donde no se utiliza opción de rechazo en el procesamiento *on-line*.

Por último, para realizar la clasificación se utilizó la Regla K-NN expresada por la siguiente ecuación:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (12)$$

4.1 Descripción de los Conjuntos de Datos

La Tabla I muestra los conjuntos de datos empleados para las pruebas, obtenidos del Repositorio SIPU (Speech and Image Processing Unit, http://cs.joensuu.fi/sipu/group.htm).

TABLA I
DESCRIPCIÓN DE LOS CONJUNTOS DE DATOS

Conjunto de Datos	D31	Joensuu	MOPI
No. Patrones	3100	6014	8589
No. *Clusters*	31	-	-
Dimensiones	2	2	2
Tamaño Archivo	49.5KB	110KB	155KB

4.2 Opción de Rechazo posterior al agrupamiento

4.2.1 Resultados Índices de Validación del Agrupamiento

El análisis de los índices de validación del agrupamiento se realizó mediante la comparación de los resultados de la implementación COR respecto a los obtenidos en el proceso SOR.

En la Fig. 1 se presentan los resultados de los índices de validación de criterio interno para los 3 CD. Primeramente, los resultados obtenidos por el índice Davies-Boulin (D-B) no presentaron una variación importante entre las pruebas SOR y las pruebas COR, predominando con valores aceptables en θ = 0.1, 0.2, 0.3.

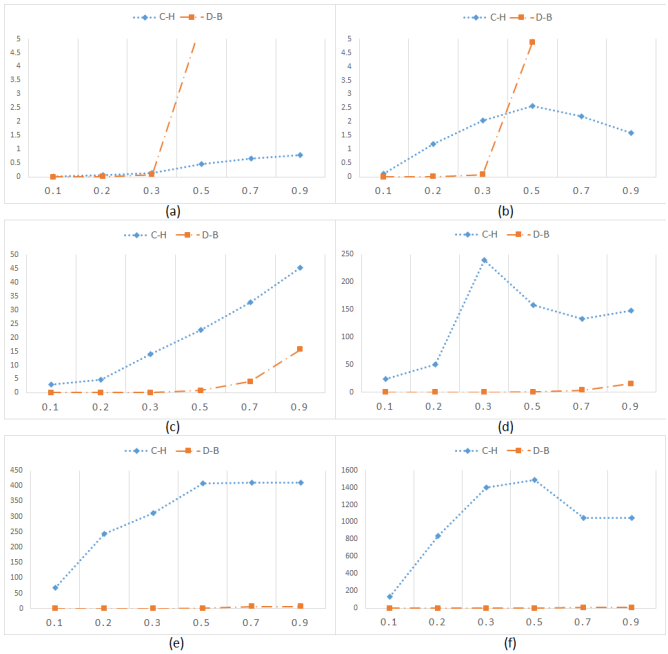


Figura 1. Índices Criterio Interno para cada valor de θ (eje x): D31 (Primer Fila), Joensuu (segunda fila), MOPI (tercera fila). Sin Opción de Rechazo (a, c, e). Con Opción de Rechazo (b, d, f).

En cuanto al índice Calinski-Harabasz (C-H), los resultados obtenidos en las pruebas COR forman una curva pronunciada sobre la cual es posible notar un pico (Fig. 1 (b, d, f)), situación que no ocurre en la estrategia SOR donde se forma una curva suave ascendente (Fig. 1 (a, c, e)). La presencia de un valor pico permite realizar un análisis más preciso del resultado obtenido para con ello identificar la mejor solución con mayor claridad, es decir el mejor agrupamiento.

Conforme a lo anterior, el mejor agrupamiento se presenta para D31 y MOPI con $\theta = 0.5$, mientras que para Joensuu con $\theta = 0.3$. Sin embargo, si se evalúan juntos ambos índices, C-H y D-B, se descartan los valores altos de θ con D-B, por lo que $\theta = 0.3$ ofrece el mejor agrupamiento para los CD D31 y Joensuu, mientras que $\theta = 0.5$ lo hace para el CD MOPI (Ver Fig. 1 (b, d, f)).

Por otro lado, tomando en cuenta que se conoce el número de grupos por los que está integrado el CD D31 (31 *Clusters*, ver Tabla I), después de evaluar los índices C-H y D-B, el mejor agrupamiento se obtiene con $\theta = 0.2$, como se observa en la Tabla II.

TABLA II
NÚMERO DE GRUPOS FORMADOS PARA EL CD D31

θ	0.1	**0.2**	0.3	0.5	0.7	0.9
No. Grupos SOR.	120	**34**	19	6	4	2
No. Grupos COR.	107	**34**	19	6	4	2

La Tabla II muestra para el agrupamiento COR, una mejora en el número de grupos obtenidos con $\theta = 0.1$ respecto al agrupamiento obtenido SOR, esto es debido a la recolocación de los patrones en los grupos con los que tienen mayor similitud, pues con un valor tan pequeño para θ los grupos tienden a ser muy pequeños y con mayor separación.

Este último análisis no es posible en los CD Joensuu y MOPI debido a que no se conoce el número de grupos que deberían obtenerse. Por lo anterior, para los CD Joensuu y MOPI, se obtiene el número de grupos que deberían ser generados conforme al procesamiento *Off-line* expuestos en [18], donde se estableció que el mejor agrupamiento para Joensuu de forma *On-line* SOR dada la similitud en cuanto a los grupos obtenidos y en base al análisis de los índices de validación se da con el valor $\theta = 0.2$ (Tabla III).

TABLA III
GRUPOS FORMADOS PARA EL CD JOENSUU

θ	0.1	0.2	**0.3**	0.5	0.7	0.9
Off-line	75	29	**15**	6	4	3
No. Grupos *On-line* SOR.	57	25	**12**	5	3	2
No. Grupos *On-line* COR.	21	18	**11**	5	3	2

Dicho lo anterior al comparar los resultados obtenidos de forma *On-line* COR y el procesamiento *Off-line*, se establece que el mejor agrupamiento se da con $\theta = 0.3$ dado que el número de grupos es más cercano entre dicha forma de procesamiento y la implementación COR, esto último coincide con el resultado del análisis de los índices de validación donde se selecciona el mismo valor de θ.

En cuanto al CD MOPI se observa un número de grupos similar en las tres formas de procesamiento, existiendo una reducción en los valores de $\theta = 0.1$ y 0.2 (Tabla IV) presentando una modificación mínima al agrupamiento SOR pero suficiente para mejorar el agrupamiento y el desempeño *on-line* COR. Lo anterior es debido a que el conjunto de datos MOPI tiene un número pequeño de grupos y por tanto cualquier modificación puede afectar de forma considerable el resultado final. Finalmente, en base al número de grupos obtenidos se podrían considerar cualquiera de los valores de $\theta = 0.3$, 0.5, 0.7 y 0.9 como una buena solución, con el apoyo de los resultados de los índices de validación los valores de $\theta = 0.3$ y 0.5 serían los que ofrecen los mejores resultados en lo que se refiere al mejor agrupamiento.

TABLA IV
GRUPOS FORMADOS PARA EL CD MOPI

θ	0.1	0.2	**0.3**	**0.5**	0.7	0.9
Off-line	19	11	**5**	**3**	2	2
No. Grupos *On-line* SOR.	14	9	**4**	**3**	2	2
No. Grupos *On-line* COR.	12	7	**4**	**3**	2	2

Finalmente, se muestra la validación del agrupamiento, del CD D31, con el método de recuento de pares partiendo del hecho de que se conoce la estructura original del CD, es decir, se sabe que está conformado por 31 *clusters* (Ver Tabla I). Los resultados de las pruebas con $\theta = 0.2$ y 0.3 se observan en la Tabla V, en donde el resultado tanto para *SS* y *DD* son altos, mientras que para *SD* y *DS* son bajos. Lo anterior indica que existe una alta similitud entre el agrupamiento COR y el generado SOR, esto deja ver que la implementación del algoritmo de opción de rechazo no afecta de forma negativa el desempeño del procesamiento *on-line*.

TABLA V
RESULTADOS RECUENTO DE PARES D31

θ	***SS***	***SD***	***DS***	***DD***
0.2	**545.3**	17	15.8	15401.5
0.3	**1006**	21.66	26	15187

El valor alto para *SS* indica que la gran mayoría de patrones se encuentran dentro del mismo *cluster* tanto en el agrupamiento obtenido en el caso SOR como en COR. Por otra parte respecto al valor obtenido para *DD*, dado que es alto, se concluye que las particiones tienen los grupos muy bien definidos, es decir no existe solapamiento entre grupos.

A partir de los valores obtenidos con el recuento de pares se realizó el cálculo de los índices de validación de criterio externo, en los cuales se puede apreciar que existe una mejoría en el valor obtenido en la implementación COR para cada uno de los índices respecto al valor en las pruebas SOR (Tabla VI).

TABLA VI
ÍNDICES DE VALIDACIÓN CRITERIO EXTERNO CD D31

Índice	**θ = 0.2**		**θ = 0.3**	
	SOR	COR	SOR	COR
Rand	0.96855	**0.997862**	0.988796	**0.996226**
ARI	0.48957	**0.726720**	0.694333	**0.717885**
Jaccard	0.33910	**0.939811**	0.852161	**0.942807**
Fowlkes & Mallows	0.50601	**0.968785**	0.919901	**0.970274**
Medida F	0.8968	**0.968747**	0.919819	**0.970266**

Como se observa tanto el índice Rand y la medida F indican que el agrupamiento COR tiene particiones muy

similares, la mejoría del resultado se observa por la recolocación de patrones en los grupos correctos. En cuanto al índice ARI, se obtuvo un valor mayor al calculado en las pruebas SOR, lo que indica que los grupos obtenidos COR son muy similares a los del CD SOR. Por su parte, el índice Jaccard indica que la mayoría de los patrones fueron bien asignados en los grupos correspondientes en la implementación COR a comparación del resultado obtenido SOR. Finalmente, el índice Fowlkes y Mallows, que no toma en cuenta los patrones del indicador *DD* aumenta su valor mostrando que la recolocación de patrones mejoró la calidad de los grupos.

4.2.1 Eliminación de patrones

Con la implementación del algoritmo de opción de rechazo se buscó mejorar la calidad de la distribución de patrones entre los grupos obtenidos del algoritmo de agrupamiento mediante la recolocación (R) o eliminación (E) de patrones. En la Tabla VII, para el CD D31 se observa que tanto la eliminación y recolocación de patrones fue equilibrada y en un bajo porcentaje, dado que el total de patrones recolocados y eliminados es menor al 8% del total de patrones del agrupamiento SOR, excepto para $\theta = 0.1$ (alrededor del 19%).

TABLA VII
COMPORTAMIENTO ALGORITMO DE RECHAZO

CD		D31		Joensuu		MOPI	
Patrones CD Original		3100		6014		8589	
Acción Algoritmo		R	E	R	E	R	E
θ	0.1	264	355	3442	25	20	3
	0.2	85	117	3442	25	8	0
	0.3	81	85	944	22	4	0
	0.5	115	106	89	65	0	0
	0.7	119	63	7	7	0	0
	0.9	60	60	15	10	0	0

De lo anterior, se puede decir que si bien existían patrones mal agrupados, el agrupamiento SOR es bueno, por ello el porcentaje de patrones modificados es bajo pero lo suficiente para mejorar el resultado del procesamiento *on-line*.

En el CD Joensuu, es posible notar que en los valores menores de θ se dio un alto porcentaje de patrones recolocados, mientras que en los valores mayores la recolocación o eliminación de patrones fue más equilibrada y en menor porcentaje. Lo anterior abre paso a la explicación del porque la mejoría en cuanto a la precisión del clasificador para este CD es más notoria en los valores donde θ es bajo y conforme θ incrementa la mejora se da en menor medida. En cuanto a los valores menores de θ (0.1, 0.2) se concluye que un poco más de 50% de los patrones estaban colocados en grupos a los cuales no pertenecían, lo cual se ve reflejado en la reducción del número de grupos obtenido por el *clustering* equilibrando los resultados de los índices de validación y con ello la mejora en el clasificador.

Finalmente en el CD MOPI se puede observar que el algoritmo de rechazo realizo una modificación mínima o nula al *clustering* pero suficiente para mejorar el agrupamiento y el desempeño del clasificador. Lo anterior es debido a que el conjunto de datos MOPI tiene un número pequeño de *clusters* y por tanto cualquier modificación puede afectar de forma considerable el resultado final.

4.2.2 Resultados del Clasificador

Respecto al rendimiento del clasificador, en todas las pruebas realizadas para los 3 CD se obtuvo una mejora en la precisión del clasificador al aplicar la opción de rechazo como se observa en la Fig. 2.

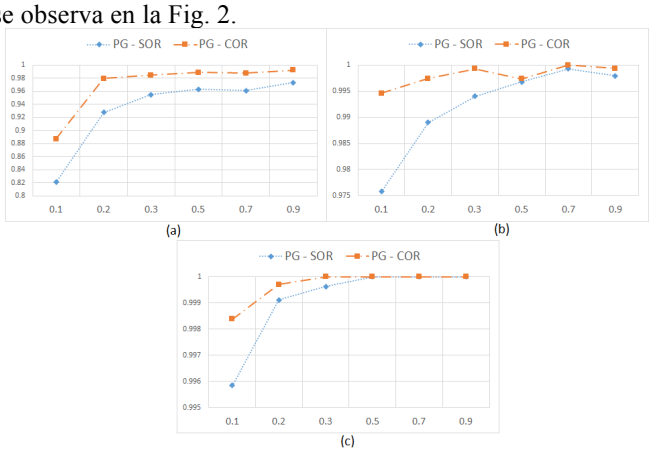


Figura 2. Precisión General obtenida para cada valor de θ (eje x) con el clasificador para cada CD Sin Opción de Rechazo y Con Opción de Rechazo. (a) D31 (b) Joensuu (c) MOPI.

Para el CD D31, la precisión en promedio es del 93% para la implementación SOR, mientras que con la implementación del algoritmo de rechazo es del 97%. En tanto que para el CD Joensuu en promedio la precisión SOR oscila entre el 98% y 99%, mientras que con la implementación COR se obtuvo una precisión mayor al 99%.

Finalmente, para el CD MOPI aunque la precisión de las pruebas SOR es muy buena, de igual forma se observó una mejora o bien se mantuvo el resultado obtenido en las pruebas COR.

En los resultados de la clasificación se logra observar que independientemente de los valores del parámetro θ, se mejoran los índices de clasificación, por lo que el resultado de la clasificación se encuentra estrechamente relacionado con la calidad del agrupamiento obtenido.

4.3 Opción de Rechazo Posterior a la Validación Cruzada

En esta sección se muestran los resultados de las pruebas con la implementación del algoritmo de rechazo posterior la etapa de validación cruzada, en donde el algoritmo se aplica únicamente al conjunto de entrenamiento que será utilizado por el clasificador. En este caso no se consideran los índices de validación ya que estos son calculados antes de la etapa de validación cruzada, y dado que el algoritmo es aplicado posterior a dicha etapa, los índices de validación permanecen igual a los resultados obtenidos en [19] y [10].

En general los resultados con esta estrategia no presentan mejoría, ya que la precisión obtenida por el clasificador disminuyo o bien en el mejor de los casos se mantuvo.

4.3.1 Resultados de Clasificación

Para los 3 CD (D31, Joensuu y MOPI), como se observa en la Fig. 3, los resultados revelaron una precisión mermada respecto al procesamiento SOR.

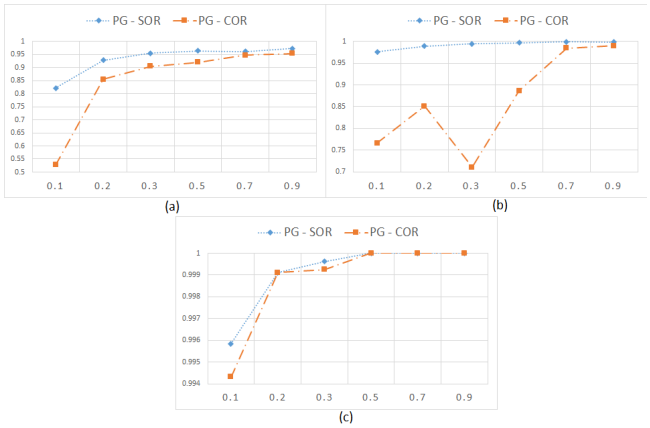


Figura 3. Precisión General obtenida para cada valor de θ (eje x) con el clasificador para cada CD Sin Opción de Rechazo y Con Opción de Rechazo. (a) D31 (b) Joensuu (c) MOPI.

Conforme a lo anterior, en la Tabla VIII se presentan los patrones recolocados (R) y eliminados (E) por algoritmo de opción de rechazo implementado después de la Validación Cruzada.

TABLA VIII
COMPORTAMIENTO ALGORITMO DE RECHAZO

CD Entrenamiento		D31		Joensuu		MOPI	
No. Patrones		2232		4330		6184	
Acción		R	E	R	E	R	E
θ	0.1	547	623	974	146	16	5
	0.2	215	270	777	510	6	3
	0.3	150	150	1534	75	2	0
	0.5	133	111	585	66	0	0
	0.7	90	93	38	49	0	0
	0.9	80	57	22	61	0	0

Para el CD D31, en los valores para θ = 0.1 y 0.2, los conjuntos de entrenamiento muestran un alto porcentaje de patrones modificados (recolocados o eliminados), 52% y 22% respectivamente. Dicha reducción genera una pérdida de información, como consecuencia muchos patrones fueron eliminados o recolocados erróneamente en otro grupo, esto último repercute tanto en la etapa de entrenamiento del clasificador así como en el resultado final del mismo. En cuanto a los valores altos de θ (0.3, 0.5, 0.7, 0.9) la precisión no disminuye debido a que, si bien existe una eliminación y recolocación de patrones en grupos distintos a los cuales pertenecen realmente, estos no afectan negativamente el funcionamiento del clasificador.

Para el CD Joensuu, las pruebas realizadas no presentan una pérdida significativa de patrones, se observa que la recolocación de los patrones se dio en mayor porcentaje en los valores de θ más pequeños (0.1, 0.2, 0.3), ya que, al tener presente un mayor porcentaje de patrones recolocados el clasificador no define adecuadamente la etiqueta para los patrones del conjunto de prueba y por ello la precisión disminuye considerablemente. Por otra parte en los valores mayores de θ donde no existe una recolocación o eliminación de patrones considerable la precisión es similar al procesamiento SOR, pero sigue siendo menor.

Finalmente para MOPI se observa que el algoritmo altero mínimamente los conjuntos de entrenamiento empleados en el clasificador, razón por la cual la precisión es muy similar o igual a la obtenida en las pruebas SOR.

4.4 Tiempo de Procesamiento

Otro aspecto importante a analizar es el tiempo requerido para el procesamiento de los conjuntos de datos. En un proceso de aprendizaje *off-line* si las condiciones del problema que se atiende cambian, es necesario actualizar el modelo de los datos (conocimiento generado), lo cual es una tarea sumamente costosa y que en ocasiones puede generar algún tipo de perdida debido a que se debe realizar nuevamente la fase de entrenamiento en su totalidad, es decir empezar de cero. Por lo cual los métodos o estrategias de aprendizaje continuo se ofrecen como alternativa de solución dado que el tiempo de procesamiento requerido teóricamente debe de ser menor. En [19] y [10] se logra apreciar que una implementación *On-line SOR* presenta un menor tiempo de procesamiento en comparación al procesamiento *Off-line*, por tanto es importante analizar si con la implementación de la opción de rechazo se mantiene la mejora respecto al tiempo requerido.

En la Fig. 4 y 5 se muestran el tiempo requerido para llevar a cabo el procesamiento para los 3 CD, posterior a la etapa de *clustering* y posterior a la etapa de validación cruzada respectivamente, donde se observa que el tiempo requerido de forma *Off-line* respecto al *On-line* SOR muestra una reducción en el tiempo donde para D31 fue de aproximadamente 47%, mientras que para Joensuu y MOPI la reducción fue de 68% y 77% respectivamente.

Por otra parte, al comparar el tiempo de procesamiento *On-line* SOR con el tiempo *On-line* COR se observa un incremento, aunque de igual forma el tiempo requerido sigue siendo menor al tiempo *Off-line*. El incremento de tiempo era esperado debido a la inclusión del algoritmo de rechazo dentro del procesamiento *On-line*.

A modo de resumen, se puede mencionar que al comparar el incremento de tiempo entre ambas propuestas se aprecia que la implementación COR posterior a la validación cruzada presenta un menor incremento en el tiempo respecto al tiempo de procesamiento *On-line* SOR, esto se debe a que el algoritmo procesa una menor cantidad de patrones debido al particionamiento del agrupamiento en la etapa de la validación cruzada.

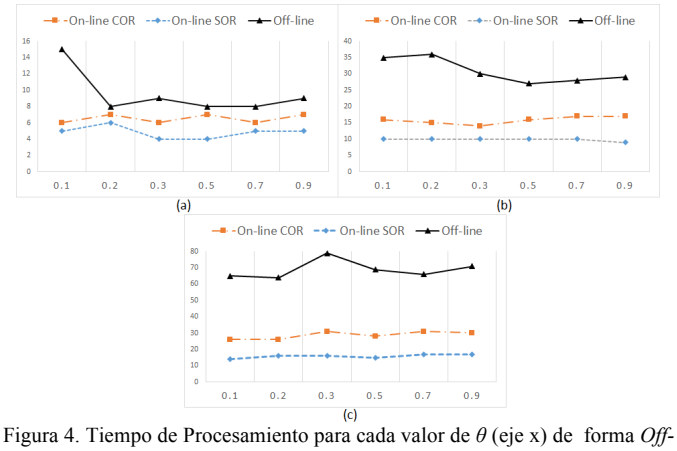


Figura 4. Tiempo de Procesamiento para cada valor de θ (eje x) de forma *Off-line*, *On-line* Sin Opción de Rechazo y *On-line* Con Opción de Rechazo posterior a la etapa de *clustering*. (a) D31 (b) Joensuu SOR (c) MOPI.

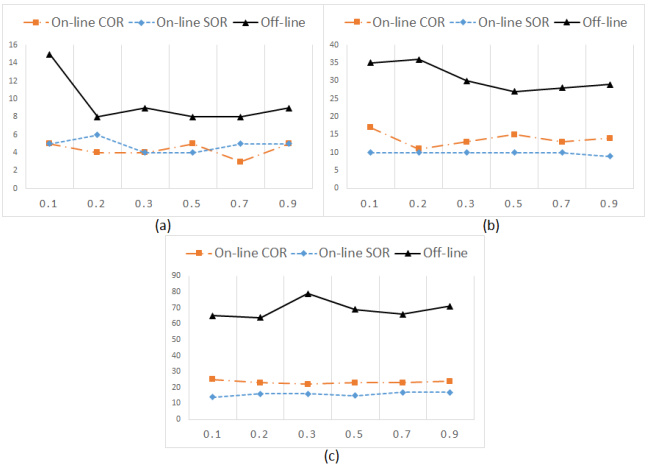


Figura 5. Tiempo de Procesamiento para cada valor de θ (eje x) de forma *Off-line*, *On-line* Sin Opción de Rechazo y *On-line* Con Opción de Rechazo posterior a la etapa de validación cruzada. (a) D31 (b) Joensuu SOR (c) MOPI.

V. CONCLUSIONES

En el presente artículo se ha analizado la implementación del algoritmo de Opción de Rechazo "Regla 2NN modificada", para el preprocesado de los datos, con miras a mejorar el procesamiento *On-line de grandes conjuntos de datos*. Se concluye que la etapa de preprocesado dentro del proceso de minería debe de ser implementada cuidadosamente, ya que si bien es cierto que el objetivo principal es mejorar el resultado de dicho proceso, si no se realiza en tiempo y forma adecuada puede derivar en un mal resultado.

Lo anterior es posible observarlo en las pruebas realizadas con la implementación del algoritmo de rechazo Posterior a la Validación Cruzada, en donde el algoritmo de rechazo es aplicado al conjunto de entrenamiento generado y por obvias razones no son considerados los patrones del conjunto de prueba. Ahora bien conforme al funcionamiento del algoritmo de rechazo, al no considerar los patrones del conjunto de prueba, aunado a que el proceso de selección de patrones para formar ambos conjuntos es aleatorio, se pierde información para determinar el accionar del mismo. En consecuencia el desempeño del clasificador se ve afectado y en el mejor de los casos se mantiene el resultado, por ende la precisión no mejora.

Por el contrario, en las pruebas con la implementación del algoritmo posterior a la ejecución del *Clustering* la precisión mejora en todos los resultados, ya que al limpiar el conjunto resultante del algoritmo de *clustering* son reasignados o eliminados patrones que durante la clasificación pudieran generar ruido, es decir se logra mejorar la calidad del agrupamiento y por ende se obtiene un conjunto de entrenamiento de mayor calidad, lo cual se ve reflejado en el resultado del clasificador.

Aunado a lo anterior, gracias a la ejecución del algoritmo de rechazo, se obtuvo una mejora en el resultado de los índices de validación para el agrupamiento, lo cual permitió determinar el mejor agrupamiento COR con mayor claridad, a diferencia de la ejecución SOR. La importancia de lo anterior radica en que dado que el resultado de la precisión no depende del valor de θ empleado sino de la forma en que se entrena al clasificador, se debe procurar trabajar con el valor de θ que ofrezca el mejor agrupamiento ya que este ofrecería un modelo de datos más parecido a uno probable en la vida real.

Otra conclusión importante a la que se llegó mediante el análisis de los datos de la primer prueba, es que para el CD D31 se obtienen muy buenos resultados con $\theta = 0.2$ y $\theta = 0.3$, mejorando los índices de validación y la precisión del clasificador. Un análisis más profundo de este resultado permitió observar que el mejor resultado para el agrupamiento se genera con el valor de $\theta = 0.2$, ya que en este caso el número de grupos formados (34) es más cercano al número real de grupos (31) esperado y aunque es mayor esto da un abanico de opciones más amplio que el ofrecido por $\theta = 0.3$.

Por otro lado, también pudo determinarse que la estructura de los grupos con $\theta = 0.2$ y 0.3 es muy diferente, pues los grupos generados con $\theta = 0.3$ son más grandes y menos definidos, por lo que es más sencillo que los patrones a recolocar sean ubicados en el grupo correcto, mientras que los grupos generados con $\theta = 0.2$ son más compactos con una frontera definida y, aunque en ellos se asignan algunos patrones de forma incorrecta, estos no merman la precisión del clasificador ni en un 0.6% conforme al resultado obtenido con $\theta = 0.3$, esto es debido a que la frontera entre los grupos está más definida y algunos de los patrones que quedan en ella son eliminados o recolocados en un grupo al que no pertenecen. De este último razonamiento una línea abierta es realizar experimentos con valores más precisos para los parámetros θ y T.

Para los conjuntos de datos Joensuu y MOPI, que cuentan con mayor número de patrones se puede concluir para Joensuu que entre más pequeño es θ (0.1, 0.2 y 0.3), es necesaria una mayor recolocación y eliminación, ya que los grupos originales son fragmentados con el algoritmo de clustering, lo que da pie a la asignación incorrecta de los patrones a los grupos. Por otra parte, los valores más altos de θ (0.5, 0.7 y 0.9) desde el algoritmo de clustering generan grupos más incluyentes al ser éstos de mayor tamaño y más parecidos a los obtenidos en el tratamiento *Off-line* de los conjuntos de datos, es por lo anterior que los agrupamientos obtenidos con los valores de $\theta = 0.5$, 0.7 y 0.9 en la clasificación entregan una precisión mayor y que entre el valor de $\theta = 0.3$ y 0.5 se ve un cambio drástico en el valor del índice C-H siendo más alto en los resultados con Opción de Rechazo gracias a la recolocación y eliminación de los patrones, esto ocasiona que los grupos, después de aplicar la Opción de Rechazo, contengan a los patrones que corresponden al grupo.

Por otra parte para el CD MOPI, si bien es cierto que es el CD con mayor cantidad de patrones, fue el CD con menor cantidad de grupos generados por el algoritmo de *clustering*, razón por la cual el algoritmo de rechazo realizo una modificación casi nula sobre el agrupamiento, ya que los grupos formados son más incluyentes y por tanto existe un menor porcentaje de patrones que pudiesen ser considerados como ruido para el clasificador, no obstante la modificación mínima realiza por el algoritmo de rechazo permitió obtener una mejora en el resultado.

Finalmente, en lo que respecta al tiempo, se anticipó que al integrar el algoritmo de Opción de Rechazo a la metodología el tiempo de procesamiento incrementaría debido a las operaciones inherentes a éste, sin embargo, se concluyó que el incremento de tiempo para los conjuntos D31 y Joensuu es

aproximadamente del 37% y 61% respecto al tiempo obtenido en las pruebas SOR, mientras que para MOPI el incremento es de un 81%, lo que equivale, en promedio a un aumento de 2 segundos para D31, 6 segundos para Joensuu y 13 segundos para MOPI. No obstante, el incremento de tiempo con la implementación COR se mantiene menor al tiempo de procesamiento *Off-line*, reduciendo en promedio para D31 un 28%, para Joensuu un 48% y para MOPI un 58%. Como es posible observar, el incremento es proporcional al número de patrones procesados conforme a la capacidad de memoria simulada, por lo tanto, en el procesamiento *On-line* mientras se procese un mayor número de patrones el tiempo incrementará proporcionalmente, pero reducirá en proporción similar al compararlo con el procesamiento *Off-line*.

AGRADECIMIENTOS

Este proyecto fue realizado gracias al apoyo recibido del proyecto SEP- PRODEP-3238 y 3834/2014/CIA de la UAEM.

REFERENCIAS

[1] D. Barber, Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012.

[2] A. Smola y S. Vishwanathan, Introduction to Machine Learning, Cambridge University Press, 2008.

[3] A. Singla y Karambir, Comparative Analysis & Evaluation of Euclidean Distance Function and Manhattan Distance Using K-Means Algorithm, *International Journal of Advanced Research in Computing Science and Software Engineering,* Vol. 2, nº 7, pp. 298-300, 2012.

[4] D. T. Larose y C. D. Larose, Data Mining and Predictive Analytics, 2nd ed., WILEY, 2015.

[5] D. L. Wilson, Asymptotic Properties of Nearest Neighbor Rules Using Edited Data, *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 1, nº 3, pp. 408-421, 1972.

[6] P. Hart, The condensed nearest neighbor rule (Corresp.), *IEEE Transactions on Information Theory,* Vol. 14, nº 3, pp. 515-516, 1968.

[7] R. Barandela, The nearest Neighbor Rule: An empirical study of its methodological aspects, Berlin, 1987.

[8] F. J. C. Bon, Técnicas no supervisadas: Métodos de Agrupamiento,» 2001. http://www.infor.uva.es/~isaac/doctorado/tema4_00-01_www.pdf.

[9] E. Rendón, I. Abundez, A. Arizmendi y E. M. Quiroz, «Internal versus External cluster validation,» *International Journal of Computers an Communications,* Vol. 5, nº 1, pp. 27-34, 2011.

[10] I. Gurrutxaga, Aportaciones a la clasificaci´on no supervisada y a su validaci´on. Aplicación a la seguridad informática, 2010. Available: https://addi.ehu.es/bitstream/10810/13910/1/2010%20Gurrutxaga%20I.pdf.

[11] Z. Ansari, M. Azeem, W. Ahmed y A.Vinaya, Quantitative Evaluation of Performance and Validity Indices for Clustering the Web Navigational Sessions, *World of Computer Science and Information Technology Journal,* Vol. 1, nº 5, pp. 217-226, 2015.

[12] S. Pandit y S. Gupta, A Comparative Study on Distance Measuring Approaches for Clustering, *International Journal of Research in Computer Science,* Vol. 2, nº 1, pp. 29-31, 2011.

[13] E. Ramirez, R. Brena, D. Magatti y F. Stella, Topic Model Validation, *Neurocomputing,* Vol. 76, nº 1, pp. 125-133, 2012.

[14] J. Santos y S. Ramos, Using a clustering similarity measure for feature selection in high dimensional data sets, de *2010 10th International Conference on Intelligent Systems Design and Applications*, Cairo, 2010.

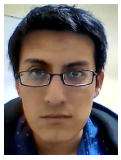
[15] W. Cheng, K. Dembczyński, E. Hüllermeier, A. Jaroszewicz y W. Waegeman, F-measure maximization in topical classificaation, *Rough Sets and Current Trends in Computing,* vol. 7413, pp. 439-446, 2012.

[16] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez y I. Perona, An extensive comparative study of cluster validity indices, *Pattern Recognition,* Vol. 46, nº 1, p. 243–256, 2013.

[17] K. Kryszczuk y P. Hurley, Estimation of the Number of Clusters Using Multiple Clustering Validity Indices, *Multiple Classifier Systems,* pp. 114-123, 2010.

[18] M. Santibáñez, R. M. Valdovinos, E. Rendón, R. Alejo y J. R. Marcial-Romero, Optimización de Recurso para el Tratamiento de Grandes Volúmenes de Datos, *Research in Computing Science Avances en Inteligencia Artificial,* Vol. 62, pp. 15-24, 2013.

[19] M. S. Sánchez, R. M. Valdovinos, A. Trueba, E. Rendón, R. Alejo y E. López, Applicability of cluster validation indexes for large data sets, de *Artificial Intelligence (MICAI), 2013 12th Mexican International Conference*, Mexico City, 2013.



Gerardo Javier Pérez Álvarez Egresado de la carrera de Ingeniería en Computación de la Facultad de Ingeniería de la Universidad Autónoma del Estado de México. Sus áreas de interés incluyen Bases de Datos, Minería de Datos y Reconocimiento de Patrones.

Mónica Santibáñez Sánchez Maestra en Ciencias de la computación, línea de acentuación en Inteligencia Artificial egresada de la Universidad Autónoma del Estado de México con intereses en el área de reconocimiento de patrones, clustering. Ingeniera en sistemas computacionales con especialidad en desarrollo web.

Rosa María Valdovinos Rosas Doctora en Ciencias Computacionales, miembro del Sistema Nacional de Investigadores. Miembro activo de la Asociación Mexicana de Inteligencia Artificial (SMIA), de la International Association of Pattern Recognition (IAPR) y de la Red Mexicana de Investigación y Desarrollo en Computación (REMIDEC).

José Raymundo Marcial Romero Doctor en Ciencias de la Computación por The University of Birmingham, UK, en The School of Computer Science. Profesor investigador de tiempo completo en la Facultad de Ingeniería de la Universidad Autónoma del Estado de México. Miembro del Sistema Nacional de Investigadores del CONACYT Nivel 1.

Marcelo Romero Huertas Doctor en Ciencias Computacionales por la Universidad de York, Inglaterra (2010) y previamente recibió el grado de Maestría en Informática con honores. Miembro del SIN del CONACYT. Profesor investigador de tiempo completo en el departamento de Ingeniería de la UAEM. Sus principales intereses en investigación son sobre procesamiento de imágenes, visión y reconocimiento de patrones. Miembro del SIN del computacional CONACYT.

Roberto Alejo Eleuterio Doctor en Ciencias Computacionales, adscrito al Instituto de Estudio Superiores de Jocotitlán, Miembro del Sistema Nacional de Investigadores del CONACYT. Los intereses en investigación se centran en la aplicación de inteligencia artificial a la solución de problemas reales.