

Evaluación del Rendimiento de un Sistema Web para Modelos de Inferencia Difusa

^aMiguel Ángel Uc Dzib, ^bVíctor Hugo Menéndez Domínguez, ^cSalvador Medina Peralta

Facultad de Matemáticas, UADY, Periférico Norte Tablaje 13615, C.P. 97110, Mérida, México.

^amgelster@gmail.com, ^bmdoming@correo.uady.mx, ^cmperalta@correo.uady.mx.

Abstract

The recent need to use mathematical models in disciplines such as Biology, Medicine, among others, has stimulated the interest in the use of new techniques. The problem is that many of these models are extremely difficult to define by a formula or exact mathematical equation and accurate due to the complexity of the system structure such as nonlinearity, uncertainty and scrambling of the actual processes. In this sense, the fuzzy inference systems allow model, complex processes or incomplete and uncertain systems. In this paper we evaluated the performance of a web system that allows to simulate models with fuzzy inference (SWID). We related the variables number of requests and processing time to compare two fuzzy algorithms through an ANCOVA (Analysis of covariance). The algorithm with the best performance used classification rules with weight factor (W).

Resumen

La reciente necesidad de utilizar modelos matemáticos en disciplinas como la Biología, Medicina, entre otras, ha estimulado el interés en el empleo de nuevas técnicas. El problema estriba en que muchos de estos modelos son extremadamente difíciles de conformar por una fórmula o ecuación matemática exacta y precisa debido a la complejidad de la estructura del sistema como por ejemplo la no linealidad, la incertidumbre y la aleatorización de los procesos reales. En este sentido, los sistemas de inferencia difusa permiten modelar procesos complejos o sistemas incompletos o inciertos. En este trabajo se evaluó el rendimiento de un sistema Web que permite simular Modelos de Inferencia Difusa (SWID, Sistema Web de Inferencia Difusa). Para ello se relacionó el número de peticiones y tiempo de procesamiento para comparar dos algoritmos difusos mediante un análisis de covarianza. El algoritmo con mejor rendimiento utiliza las reglas de clasificación con un factor de peso (W).

Keywords and phrases : SWID, Inferencia Difusa, Fuzzy, Covarianza, Web.

2015 *Mathematics Subject Classification* 68Q60,03B52,03b70,68T30.

1. Introducción

Internet ha tenido una marcada evolución en los últimos años, permitiendo el acceso e intercambio de información de manera ágil, con características de flexibilidad en la tecnología de acceso y con capacidades de integración a nivel de servicios [1].

Existen disciplinas como la Biología, Medicina, Psicología, Educación, entre otras, donde se requiere ejecutar diferentes modelos matemáticos en la computadora, sin embargo, debido a la complejidad en su

ejecución se requiere generar ambientes fáciles de utilizar para los usuarios. El rendimiento o el tiempo destinado en la generación de estos ambientes utilizando herramientas Web permite decidir si es viable desarrollarlos a través de una arquitectura propuesta. La modelación utilizando lógica difusa o borrosa es una propuesta relevante dada la posibilidad de representar la incertidumbre y la aleatorización presente en los problemas existentes en las áreas antes mencionadas. Esta modelación difusa tiene impacto práctico en el control, la predicción y la deducción [2].

Un sistema de inferencia difusa emplea reglas difusas “si-entonces” que pueden modelar los aspectos cualitativos del conocimiento humano y procesos de razonamiento sin emplear análisis cuantitativos precisos. En 1974 el profesor Mamdani experimentó con éxito un controlador difuso en una máquina de vapor, pero la primera implantación real de un controlador de este tipo fue realizada en 1980 por F. L. Smidth & Co. en una planta cementera en Dinamarca [9]. En 1983, por primera vez, en Japón Fuji aplicó lógica borrosa para el control de inyección química para plantas depuradoras de agua. En 1987 la empresa OMRON desarrolla los primeros controladores borrosos comerciales con el profesor Yamakawa. A partir de ese momento, el control difuso ha sido aplicado con éxito en muy diversas ramas tecnológicas, por ejemplo la metalurgia, los robots de fabricación, controles de maniobra de aviones, ascensores o trenes (tren-metro de Sendai, Japón, 1987), sensores, imagen y sonido (sistema de estabilización de imagen en cámaras fotográficas y de video Sony, Sanyo, Canon), electrodomésticos (lavadoras de Panasonic o Bosch, aire acondicionado Mitsubishi, rice cooker), automoción (sistemas de ABS de Mazda o Nissan, cambio automático de Renault, control automático de velocidad, climatizadores) y una larga lista de aplicaciones comerciales [9].

En este trabajo se evaluó el rendimiento de un sistema Web que permite simular Modelos de Inferencia Difusa para la modelación de datos, el cual utiliza el lenguaje de programación R [3] como motor de inferencia estadístico. Se explica la relación entre el número de peticiones y tiempo de procesamiento para comparar dos algoritmos difusos mediante un análisis de covarianza.

En la sección 2 se describe el desarrollo de la aplicación denominada SWID (Sistema Web de Inferencia Difusa) en términos de la estructura y rendimiento, así como su relación con la lógica difusa. La sección 3 describe el proceso de experimentación de los datos, recursos de hardware, software y modelo estadístico utilizado. La sección 4 presenta el análisis e interpretación de los resultados del experimento. En la sección 5 se presenta las conclusiones y trabajo futuro del proyecto.

2. Desarrollo de la aplicación

En el desarrollo de las aplicaciones Web se han establecido conjuntos de directrices que generalmente se enfocan en controlar la estructura de los contenidos para una fácil lectura y comprensión de los usuarios. Dos técnicas muy socorridas para mejorar el desempeño de una aplicación Web son la compresión de información y el uso de caché en el cliente y el servidor Web [10]. Los beneficios son evidentes, el tiempo de carga y la transferencia de datos para los usuarios se realizan de una manera rápida.

Entre los métodos de compresión se encuentra el gzip. Durante mucho tiempo fue el formato de compresión más popular. Como su nombre indica emplea el mismo algoritmo utilizado en la herramienta gzip de UNIX. Al utilizarse en el protocolo HTTP se espera una relación de compresión de aproximadamente 1:3 o 1:4 para HTML, pero es altamente dependiente de la naturaleza del texto (en particular cuando los patrones se repiten). El almacenamiento en caché es una parte importante del Internet moderno, pero también es un tema rodeado de mucha confusión y malos entendidos. Sin embargo, el concepto de caché es importante para garantizar la fluidez de la Web. El almacenamiento en caché es un término amplio, pero por lo general se refiere a la conservación de los recursos Web (documentos HTML, imágenes, etc.) en una ubicación temporal para mejorar el rendimiento de una aplicación. Puede ser implementado por la mayoría de los navegadores y servidores Web en diferentes sistemas operativos [11].

Para la implementación de SWID se utilizó la Arquitectura SaaS (software como servicio). El comportamiento dinámico de SaaS puede variar mucho dependiendo de la protocolos, estándares y enfoques arquitectónicos elegidos [12]. Este patrón puede ser implementado en muchos niveles de complejidad, desde un servicio Web seguro a una interfaz de usuario simple, como una página Web en un navegador. El usuario interactúa con el servicio, que a su vez invoca a la funcionalidad básica. Las respuestas son adecuadamente

dirigidas de vuelta al usuario según se requiera. Un proveedor de software de una aplicación SaaS también puede reaccionar rápidamente a los errores en el sistema. El administrador puede supervisar todos los usuarios de forma concurrente para detectar fallas en el software que pueden requerir atención inmediata y pueda ser capaz de solucionarlos antes de que la mayoría de los usuarios se den cuenta. La arquitectura SaaS puede ser especializada mediante el uso de algoritmos avanzados de computación para realizar tareas de razonamiento tales como la inferencia. Estos sistemas avanzados algún día serán capaces de utilizar las habilidades cognitivas para reconocer y actuar sobre ciertos patrones de eventos [12].

Se seleccionó HTML, PHP y AJAX como tecnologías base para SWID, el cuál utiliza el lenguaje de programación R [3] como motor de inferencia estadístico. R es un lenguaje de programación eficiente en el análisis de datos estadísticos y la simulación de modelos matemáticos, además que se puede conectar con MySQL, SQLite, MongoDB y Hadoop para actividades de almacenamiento de datos [15].

SWID es un sistema Web que permite generar modelos de inferencia difusa en áreas como la Biología, Medicina, Psicología, Educación, entre otras. SWID ofrece las funciones de gestión de datos, funciones de pertenencia y análisis de datos para dos tipos de algoritmos de clasificación difusa, el primero implementa la técnica Chi's (CHI) [4] y el segundo utiliza un factor de peso (W) [5].

La figura 1 muestra la arquitectura de software como servicio propuesta para el SWID.

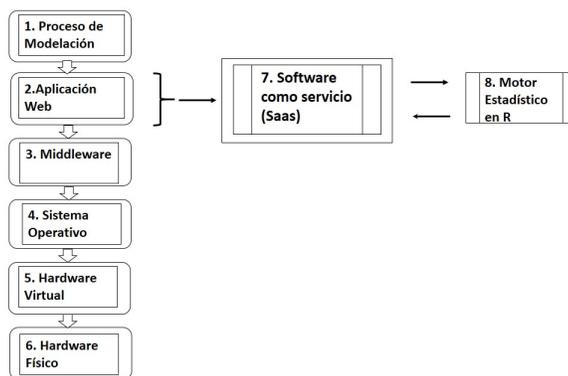


Figura 1: *Arquitectura SaaS (arquitectura de software como servicio), 1.- Proceso de modelación, 2.- Aplicación Web, 3.- Middleware, 4.-Sistema Operativo, 5.-Hardware Virtual, 6.- Hardware Físico, 7.- Software como Servicio, 8.- Motor Estadístico en R.*

La lógica difusa es una extensión de la lógica matemática basada en la teoría de conjuntos difusos y sistemas lógicos de varios valores infinitos. La lógica difusa se ha convertido en un campo importante de la investigación y el desarrollo en muchas disciplinas como las matemáticas, la lógica, la inteligencia artificial y la filosofía desde 1965 [7]. Las posibilidades de aplicación de los algoritmos evolutivos en el campo de la lógica difusa están documentadas por una serie de publicaciones recientes de investigación, que a grandes rasgos permiten optimizar las funciones de pertenencia de los conjuntos borrosos, además de optimizar el aprendizaje automático de las reglas difusas [8].

Las funciones de pertenencia proporcionan la caracterización de conjuntos difusos mediante el establecimiento de una conexión entre términos lingüísticos (como "lenta", "media", "rápida" para una variable de velocidad) y los valores numéricos precisos de variables en un sistema físico. Una función de pertenencia difusa se aproxima a la confianza con la que un valor numérico es descrito por un término lingüístico. La correcta elección de las funciones de pertenencia no es en absoluto trivial pero juega un papel crucial en el éxito de una aplicación [8].

Los sistemas de inferencia difusa emplean diversas técnicas para la generación de reglas o modelos, una de ellas es la tabla de repertorio difuso. La tabla de repertorio difuso es una matriz rectangular con elementos (columnas) y constructos (filas). Cada intersección fila-columna contiene una razón, la cual consiste de un conjunto de funciones de pertenencia trapezoidales o triangulares, definidas como $f : u \rightarrow [0, 1]$ con cuatro parámetros (ver Ec. 2.1) que muestran cómo un usuario aplica un constructo dado a un elemento en parti-

cular [16]. Las funciones trapezoidales o triangulares, extienden las tablas de repertorio difuso para formar restricciones, las cuales pueden ser representadas en una rejilla para darle mayor poder a la representación, mientras se le permite al usuario expresar su conocimiento sin restricciones.

$$f(u; a, b, c, d) = \begin{cases} 0 & u < a \\ \frac{(u-a)}{(b-a)} & a \leq u < b \\ 1 & b \leq u \leq c \\ \frac{(d-u)}{(d-c)} & c < u \leq d \\ 0 & u > d \end{cases} \quad (2.1)$$

Las reglas difusas SI - ENTONCES para c patrones de clasificación con n atributos tiene la notación siguiente [2] (ver 2.2):

$$\begin{aligned} \text{Una Regla } R_j : \text{ Si } x_1 \text{ es } A_{j1} \text{ y } \dots \text{ y } x_n \text{ es } A_{jn} \\ \text{entonces } C_j, j = 1, 2, \dots, N \end{aligned} \quad (2.2)$$

donde :

$$\begin{array}{ll} x = (x_1, \dots, x_n) & \text{vector patrón de dimensión } n \\ A_{ji} & \text{valor de la lingüística antecedente } (i = 1, 2, \dots, n) \\ C_j & \text{clase consecuente} \\ N & \text{número de reglas difusas del tipo SI - ENTONCES} \end{array}$$

Para problemas con función de aproximación (ver 2.3), las reglas difusas SI - ENTONCES con números reales como valores consecuentes utiliza la siguiente notación:

$$\begin{aligned} \text{Una Regla } R_j : \text{ Si } x_1 \text{ es } A_{j1} \text{ y } \dots \text{ y } x_n \text{ es } A_{jn} \\ \text{entonces } y \text{ es } r_j, j = 1, 2, \dots, N \end{aligned} \quad (2.3)$$

donde y es la variable de salida y r_j es el consecuente número real.

Para estimar el valor de salida \hat{y} de un vector de entrada $x = (x_1, \dots, x_n)$ se calcula como el promedio de los pesos utilizando la siguiente fórmula (ver 2.4):

$$\hat{y} = \frac{\sum_{j=1}^N \mu_j(x) \cdot r_j}{\sum_{j=1}^N \mu_j(x)} \quad (2.4)$$

donde r_j representa el consecuente número real y $\mu_j(x)$ es el grado de compatibilidad de la regla R_j . El grado de compatibilidad $\mu_j(x)$ se calcula como el operador producto $\mu_j(x) = \mu_{j1}(x_1) \times \dots \times \mu_{jn}(x_n)$ donde $\mu_{ji}(\cdot)$ es la función de pertenencia de la lingüística antecedente A_{ji} ($i = 1, \dots, n$).

Si se considera el peso w_j de cada regla en (2.3), la estructura queda de la siguiente manera (ver 2.5 y 2.6):

$$\begin{aligned} \text{Una Regla } R_j : \text{ Si } x_1 \text{ es } A_{j1} \text{ y } \dots \text{ y } x_n \text{ es } A_{jn} \\ \text{entonces } y \text{ es } r_j \text{ con } w_j, j = 1, 2, \dots, N \end{aligned} \quad (2.5)$$

La ecuación 2.4 queda de la siguiente manera :

$$\hat{y} = \frac{\sum_{j=1}^N \mu_j(x) \cdot r_j \cdot w_j}{\sum_{j=1}^N \mu_j(x)} \quad (2.6)$$

Se asume que el peso w_j afecta solo la parte consecuente r_j [14].

Los sistemas difusos se han utilizado gracias a su capacidad para resolver con éxito una amplia gama de problemas en diferentes campos de aplicación. Sin embargo su replicación y aplicación requiere un alto nivel de conocimiento y experiencia en computación. Es así, que si bien algunos investigadores publican el código fuente de sus soluciones software, su uso queda limitado a expertos en el ámbito de computación pues investigadores de otras áreas como Biología o Sociales carecen de los conocimientos para la adecuación de dichas soluciones. En los últimos años, el software en la modelación de sistemas difusos se ha desarrollado con el fin de facilitar el uso, entre los cuales podemos encontrar [17]:

1. En el área de control difuso está jfuzzy-logic, warpstar2 y asafes2.
2. Para resolver problemas específicos en la toma de decisiones los programas más conocidos son fmcgds – decider, visualfcm, xfuzzy entre otros.
3. En el contexto de la biomedicina hay muchas aplicaciones tales como msclust, fuzzy spike sorting y pinti.
4. En el área educativa existen bibliotecas para las tareas de enseñanza como coala, cavus, eriskgame, virtuallab, feup, edurobot, firs-trainer, fuzzy ramset, simulink y mntool.
5. Como servicios web está fws (fuzzy Web system) que es una biblioteca que utiliza un motor de inferencia difuso basado en el contexto de casandra (sistema de simulación de adaptación consciente para aplicaciones en numerosas áreas) [18].

Por otro lado, muchas bases de datos y herramientas de análisis de datos se han creado para la investigación bioinformática. Generalmente se combinan para ser utilizados en las tareas de recuperación y de análisis de datos por algunos biólogos y científicos en bioinformática. SWID es una propuesta de software que puede ser usado como herramienta de investigación (ver figura 2) no solo en el área de la bioinformática sino también en otras disciplinas como Medicina, Psicología, Educación, entre otras [10].

Figura 2: Interfaz Web para SWID.

SWID tiene cuatro características particulares:

1. Provee una interfaz Web para generar modelos de inferencia difusa.
2. Controla la estructura de los contenidos para una fácil lectura y comprensión de los usuarios.
3. Utiliza sistemas difusos basado en reglas para tareas de clasificación y regresión.
4. Genera funciones de pertenencia personalizados y sus gráficas asociadas.

Con todo ello, SWID se convierte en una alternativa interesante para el investigador que desea utilizar los sistemas de inferencia difusa como técnica para la modelación de datos, sin considerar las complejidades propias del uso de técnicas computacionales.

3. Diseño Experimental

Con el propósito de comparar el rendimiento entre los algoritmos, se utilizaron datos de un muestreo de los laboratorios de investigación marina de la universidad de Tasmania en Australia [13], se capturaron un total de 4,117 moluscos del género *Haliotis* (abulones). Las variables morfológicas de entrada fueron longitud, diámetro, altura, peso total, peso sin concha, peso de la víscera, peso de la concha y el número de anillos que fue la variable de salida (Tabla 1).

Tabla 1: Variables morfológicas.

Nombre	Descripción	Unidades	Mínimo	Máximo
Longitud	continua	mm	0.075	0.815
Diámetro	continua	mm	0.055	0.650
Altura	continua	mm	0.000	1.130
Peso total	continua	gr	0.002	2.826
Peso sin concha	continua	gr	0.001	1.488
Peso de la víscera	continua	gr	0.0005	0.760
Peso de la concha	continua	gr	0.0015	1.005
Anillos	discreta	número de anillos	1	29

Todos los experimentos se ejecutaron en una computadora Toshiba satélite con procesador AMD A6, 4 GB de RAM y sistema operativo Debian 8 en la cual se instaló un servidor Apache 2.1, ApacheBench 2.3, PHP 5.0, R 3.1.3 y la aplicación SWID. Se ejecutaron 100, 150, 200, 250, 300, 350 y 400 peticiones de manera aleatoria con un nivel de concurrencia de 10 utilizando la herramienta ab (Apache HTTP server benchmarking tool 2.3, Figura 3) [6]. Se relacionó la variable número de peticiones (covariable) con el tiempo total de ejecución, y se utilizó un análisis de covarianza para verificar diferencias entre dos tipos de algoritmos con respecto al tiempo total de ejecución, el primero implementa las reglas de clasificación usando la técnica Chi's (FRBCS.CHI) [4] y el segundo las reglas de clasificación con un factor de peso (FRBCS.W) [5].

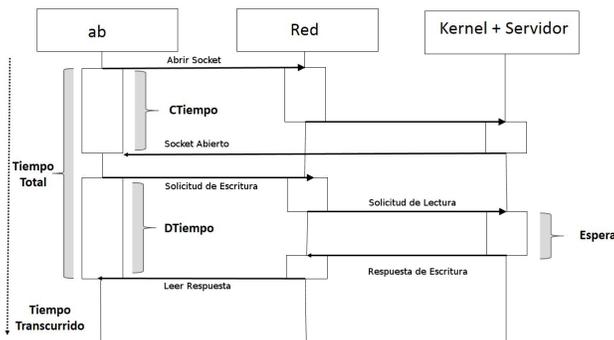


Figura 3: Diagrama de tiempos de procesamiento de ab, *CTiempo* : tiempo que establece la conexión y podemos enviar / escribir nuestra petición de servidor (que es tiempo de conexión a la red + tiempo en aplicación), *DTiempo* : cantidad de tiempo que se refiere a (tiempo de escribir) + (tiempo para la elaboración del servidor) + (tiempo para leer la respuesta).

Con respecto al tiempo total de ejecución, los parámetros de entrada para los dos algoritmos fueron 7 etiquetas y una función de pertenencia trapezoidal. Los resultados de las pruebas estadísticas fueron considerados significativos si $P < 0.05$ y el paquete estadístico utilizado fue R 3.1.3 [3]. El Modelo estadístico (ver Ec. 3.1) empleado en un análisis de covarianza es el siguiente:

$$y_{ij} = \mu + \tau_i + \beta(x_{ij} - \bar{x}_{..}) + \varepsilon_{ij} \quad (3.1)$$

en donde y_{ij} es el j -ésimo tiempo de procesamiento bajo el i -ésimo algoritmo o nivel del único factor, x_{ij} es el valor del número de peticiones llamada covariable o variable concomitante correspondiente a y_{ij} (ij -ésimo ensayo), $\bar{x}_{..}$ es el promedio de las x_{ij} , τ_i es el efecto del i -ésimo algoritmo, β es el coeficiente de regresión lineal que indica la dependencia entre y_{ij} y x_{ij} , ε_{ij} es un componente de error aleatorio. Suposiciones: Los errores ε_{ij} son NID $(0, \sigma^2)$, la pendiente β es diferente de cero, la relación real entre x_{ij} y y_{ij} es lineal, la variable concomitante x_{ij} no depende de los tratamientos.

4. Análisis de Resultados

En la tabla 2 se observan las etiquetas lingüísticas para cada variable de entrada del sistema difuso a generar para el abulón macho:

Tabla 2: *Etiquetas.*

EExtra pequeña	Extra pequeña	Pequeña	Mediana	Grande	Extra Grande	EExtra Grande
XXS	XS	S	M	L	XL	XXL

La tabla difusa (ver Ec. 2.1 y tabla 3) que se obtuvo al ejecutar el entrenamiento del modelo para la variable longitud es la siguiente :

Tabla 3: *Tabla difusa de Longitud.*

Parámetros \ Etiquetas	Etiquetas						
	XXS	XS	S	M	L	XL	XXL
a	0	0.1277	0.2555	0.3833	0.5111	0.6388	0.7777
b	0.1111	0.2388	0.3666	0.4944	0.6222	0.7500	0.8888
c	0.2222	0.2944	0.4222	0.55	0.6777	0.8055	1
d	NA	0.4055	0.5333	0.6611	0.7888	0.9166	NA

La tabla 4 muestra la predicción (ver Ec. 2.4) del número de anillos con respecto a la Longitud, Diámetro, Altura, Peso total, Peso sin concha, Peso de la víscera y Peso de la cáscara.

Tabla 4: *Tabla de predicción.*

Regla	Anillos
SI Longitud es S y Diámetro es S y Altura es XXS y Peso total es XXS y Peso sin concha es XXS y Peso de la víscera es XXS y Peso de la cáscara es XXS	26

La figura 4(a) Longitud, 4(b) Diámetro, 4(c) Altura, 4(d) Peso total, 4(e) Peso sin concha, 4(f) Peso de la víscera y 4(g) Peso de la cáscara representa cada variable medida del abulón con sus dominios difusos.

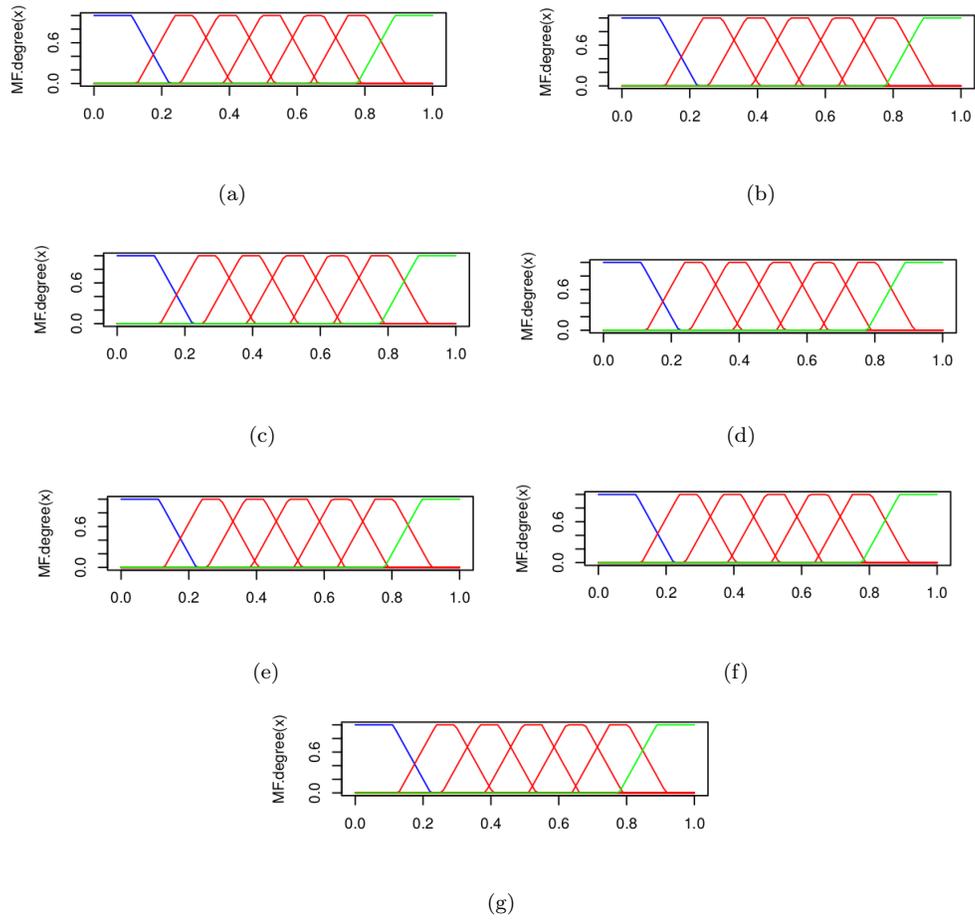


Figura 4: Ajuste de los límites de clasificación utilizando funciones de pertenencia trapezoidal.

La figura 5 muestra el número de peticiones (100, 150, 200, 250, 300, 350, 400) y el tiempo de la prueba realizada en milisegundos. Puede notarse que a mayor número de peticiones el tiempo de ejecución se incrementa para los dos tipos de algoritmos.

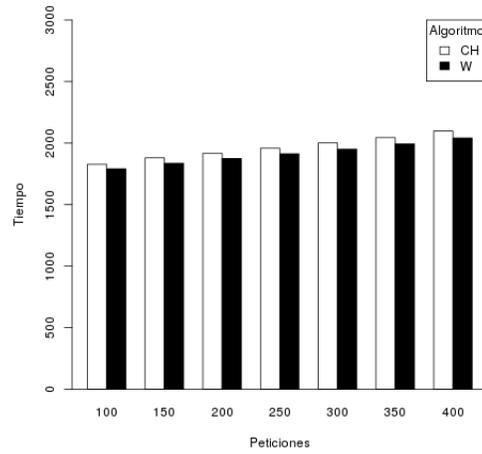


Figura 5: Número de peticiones contra el tiempo.

Se cumplió tanto el supuesto de normalidad (prueba de bondad de ajuste de Kolmogorov Smirnov : $D = 0.0516$, $P = 0.0805$, figura 6), como el de igualdad de varianzas (figura 7) para los residuos.

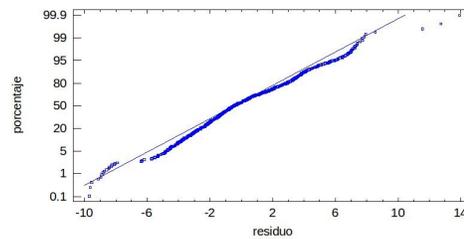


Figura 6: Gráfica de probabilidad normal.

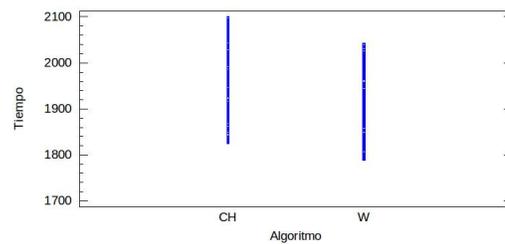


Figura 7: Gráfica de residuos vs algoritmos.

Del análisis de covarianza se determinó una relación lineal del tiempo de procesamiento con el número de peticiones ($F = 245,577.53$, $P < 0.0001$) y diferencias significativas entre algoritmos respecto al tiempo de procesamiento ($F = 24,246.08$, $P < 0.0001$, figura 8).

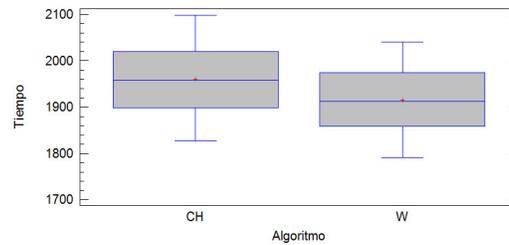


Figura 8: *Tiempos de procesamiento por algoritmo.*

5. Conclusiones y trabajo futuro

El experimento muestra que el sistema SWID es útil para modelar procesos de inferencia difusa de manera eficiente ya que los tiempos medios de procesamiento fueron de 1960.11 y 1914.98 milisegundos para los algoritmos CH y W respectivamente; el que resultó con mejor rendimiento fue el algoritmo W. Los resultados resaltan la importancia de una adecuada arquitectura y configuración del servidor Web para garantizar una mejor transferencia de información. Cabe señalar que el lenguaje de programación R es útil como motor de inferencia difusa en tecnologías web con PHP. El uso del servidor Web Apache y la herramienta ab facilitó en gran medida la evaluación de SWID. Como trabajo futuro se propone evaluar el rendimiento de SWID variando el número de etiquetas y el tipo de función de pertenencia (la gaussiana, la triangular), así como proponer otros tipos de algoritmos en el área de la lógica difusa.

Referencias

- [1] P. Lago, R. Farenhorst, P. Avgeriou, R.C. de Boer, V. Clerc, A. Jansen, and H. Van Vliet. *The Griffin Collaborative Virtual Community for Architectural Knowledge Management*. in Collaborative Software Engineering, Eds. Springer, 2010, pp. 195-217.
- [2] T. Takagi and M. Sugeno. *Fuzzy identification of systems and its applications to modeling and control*. IEEE Transactions on Systems, Man and Cybernetics, 116-132, 1985.
- [3] I. Ross. *R : A Language for Data Analysis and Graphics*. J. Comput. Graph. Stat., 299-314, 1996.
- [4] Z. Chi, H. Yan, and T. Pham. *Fuzzy algorithms with applications to image processing and pattern recognition*. World Scientific, Singapore, 1996.
- [5] H. Ishibuchi and T. Nakashima. *Effect of Rule Weights in Fuzzy Rule-Based*. IEEE Transactions on Fuzzy Systems, 506 - 515, 2001.
- [6] Apache Software Foundation. *Apache HTTP Server Reference Manual for Apache version 2.2.17*. Network Theory Ltd., EUA, 2010.
- [7] Z. Sun and G. Finnie. *A Fuzzy Logic Approach to Experience Based Reasoning*. International Journal of Intelligent Systems, Vol. 22, 867-889 , 2007.
- [8] T. Back and F. Kursawe. *Evolutionary Algorithms for Fuzzy Logic: A Brief Overview*. World Scientific Publishing Co. Pte. Ltd., 1995.
- [9] J. A. Olivas. *La lógica borrosa y sus aplicaciones*. Universidad de Castilla La Mancha, 2007.

- [10] M. Uc, V. Menéndez, y S. Medina. *Optimización del desempeño de un Sistema Web para Modelos de Inferencia Difusa mediante Técnicas no invasivas*. Tecnología educativa CONAIC, 2395-9061, 27–31, 2016.
- [11] P. Smith. *Professional Website Performance : Optimizing the Front End and the Back End*. John Wiley & Sons, Inc., 2013.
- [12] G. James , H. Dion, and N. Duane. *Web 2.0 Architectures*. O’Reilly Media, 2009.
- [13] M. Lichman. *UCI Machine Learning Repository*. <http://archive.ics.uci.edu/ml>, University of California, School of Information and Computer Science.
- [14] D. Nauck and R. Kruse. *How the learning of rule weights affects the interpretability of fuzzy systems*. in Proc. 7th IEEE Int. Conf. Fuzzy Systems, Anchorage, AK, May 4–9, 1998, pp. 1235–1240.
- [15] V. Prajapati. *Big Data Analytics with R and Hadoop*. Packt Publishing, ISBN 978-1-78216-328-2, 2013.
- [16] J.J. Castro, J.L. Castro, and J.M. Zurita. *Fuzzy Repertory Table: A Method for Acquiring Knowledge About Input Variables to Machine Learning Algorithm*. IEEE Transactions on Fuzzy Systems, 123-139, 2004.
- [17] J. Alcalá and J.M Alonso. *A Survey of Fuzzy Systems Software: Taxonomy, Current Research Trends and Prospects*. IEEE Transactions on Fuzzy Systems, 2016.
- [18] A. Javor and A. Fur. *Simulation on the web with distributed models and intelligent agents*. Simulation-Transactions of the Society for Modeling and Simulation International, vol. 88, no. 9, pp. 1080–1092, 2012.