Advances in Artificial Intelligence

Research in Computing Science

Series Editorial Board

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 113,** septiembre 2016. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. : 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor responsable: *Grigori Sidorov, RFC SIGR651028L69*

Research in Computing Science is published by the Center for Computing Research of IPN. **Volume 113**, September 2016. Printing 500. The authors are responsible for the contents of their articles. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Advances in Artificial Intelligence

Grigori Sidorov (ed.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2016

ISSN: 1870-4069

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.rcs.cic.ipn.mx
http://www.ipn.mx
http://www.cic.ipn.mx

Indexed in LATINDEX, DBLP and Periodica

Printing: 500

Printed in Mexico

Editorial

This volume of the journal "Research in Computing Science" contains selected papers related to machine learning, automatic control and their applications. The papers were carefully chosen by the editorial board on the basis of the at least two reviews by the members of the reviewing committee or additional reviewers. The reviewers took into account the originality, scientific contribution to the field, soundness and technical quality of the papers. It is worth noting that various papers for this special issue were rejected.

The volume contains 14 papers, which treat various aspect of modelling using Artificial Intelligence techniques in different domains. Several papers deal with medical applications, other areas are wind turbine, soccer modelling, social activities, PID controller, robotics, among others.

I would like to thank Mexican Society for Artificial Intelligence (Sociedad Mexicana de Inteligencia Artificial) and COMIA 2016.

The entire submission, reviewing, and selection process, as well as preparation of the proceedings, were supported for free by the EasyChair system (www.easychair.org).

Grigori Sidorov
Guest Editor
Instituto Politécnico Naiconal,
CIC-IPN, México

September 2016

Table of Contents

Modelo difuso para la predicción de casos de obesidad empleando el árbol GFID3 generalizado

Christian Suca, André Córdova, Abel Condori, Jordy Cayra

Universidad Nacional de San Agustín,
Escuela Profesional de Ingeniería de Sistemas, Arequipa,
Perú

{andrepacord, csucavelando, perezz}@gmail.com, acondoricas@unsa.edu.pe

Resumen. La obesidad es la base de muchas enfermedades crónicas importantes, y por lo tanto es uno de los mayores problemas de salud en el mundo, siendo el IMC el principal indicador empleado para la predicción, sin embargo, el porcentaje de grasa corporal es uno de los factores asociados con problemas de obesidad, siendo estos porcentajes distintos para hombres y mujeres. El objetivo de este artículo es desarrollar un modelo novel difuso clasificado para hombres y mujeres para la predicción de casos de obesidad en personas de 6-17 años. El presente trabajo estudió los factores e indicadores empleados como medidas de predicción de casos de obesidad y desarrolló un modelo difuso que puede ser utilizado como un índice de predicción de obesidad en el campo médico utilizando percentiles de peso, altura y el porcentaje de grasa corporal (BF). Los pasos involucrados en este estudio son: una revisión de los factores de obesidad infantil, colección Pre-procesamiento y clasificación de los datos, fuzzificación de las entradas y la generación de reglas óptimas, basadas en el árbol GFID3. Los resultados obtenidos demuestran que el modelo difuso desarrollado obtiene una exactitud 83.65% para hombres y 76.13% en mujeres para la predicción de casos de obesidad.

Palabras clave: Obesidad, árbol GFID3, lógica difusa, IMC.

Fuzzy Model for Prediction Obesity Cases Using the Generalized GFID3 Tree

Abstract. Obesity is the cause of many major chronic diseases, and therefore is one of the biggest health problems in the world, being the IMC the primary indicator used for prediction, however, the percentage of body fat is one of the factors associated with obesity problems, these percentages being different for men and women. The aim of this paper is to develop a new classified novel fuzzy model for men and women for the prediction of cases of obesity in people aged 6-17 years. This paper studied the factors and indicators used as measures of prediction cases of obesity and developed a fuzzy model which can be used as a predictor of obesity in the medical field using percentiles weight, height and percentage of body fat (BF). The steps involved in this study are: a review of the

factors of childhood obesity, Pre-processing collection and classification of data fuzzification of inputs and generating optimal rules, based on the tree GFID3. The results show that the developed fuzzy model accuracy gets 83.65% and 76.13% for men and women for predicting cases of obesity.

Keywords: Obesity, GFID3 tree, fuzzy logic, IMC.

1. Introducción

La condición clínica conocida como obesidad actualmente asume la característica de epidemia universal, esta enfermedad es causa importante de muerte y enfermedades relacionadas (también denominadas co-morbilidades), por otra parte se relaciona directamente y de forma proporcional a la magnitud de la condición clínica de la Obesidad. La OMS [6] indica que el número de niños y lactantes que padecen sobrepeso aumento de 32 millones a 42 millones en el 2013, la OMS establece que de acuerdo a la tendencia posiblemente este número crezca hasta 70 millones para el 2025. La definición de obesidad, medición, clasificación y tratamiento no se determinan fácilmente, encontrar un sistema para evaluar y clasificar el nivel de Obesidad, sobre todo si se refiere a los riesgos o la prescripción de los tratamientos es de gran interés clínico [1]. El exceso de peso puede provenir de músculo, hueso, grasa y / o agua en el cuerpo, la presencia de exceso de grasa en el tronco o en el abdomen fuera de proporción con la grasa corporal total es un predictor independiente de los factores de riesgo y morbilidad [19]. Hoy en día se realizan múltiples investigaciones para clasificar a una persona en condición de sobrepeso, obesidad, etc. utilizando diversos parámetros antropométricos, cuando el interés es determinar la mejor indicación terapéutica y el tratamiento, así como el mejor mecanismo apropiado para el análisis de datos, la búsqueda de un método más preciso para evaluar la obesidad y en consecuencia el mejor tratamiento [1]; por este motivo surge la pregunta, es posible la aplicación y utilización de un modelo difuso basado en árboles de decisión que pueda ser utilizado como un índice de predicción de casos de obesidad empleando un conjunto de registros médicos de crecimiento de personas entre 6-17.

En esta investigación, se desarrolló un modelo difuso para la predicción de casos de obesidad. Este enfoque novel combina las técnicas de árboles de decisión difuso ID3 y la lógica difusa para la generación de un índice basado en un conjunto de reglas así como un modelo para la predicción de casos de obesidad tomando como referencia los percentiles de edad, peso y BF y que pueden ser tomados como ayuda en el campo médico. Este modelo está clasificado tanto para hombres como para mujeres, tomando como referencia el hecho de que las medidas y rangos, como el porcentaje de grasa tienen rangos distintos en hombres y mujeres [1], por lo tanto el modelo clasificado es capaz de predecir si una persona es propensa a casos de obesidad con mayor precisión y ofrecen una alternativa de ayuda en el campo médico. Para relacionar las diferentes variables de nuestra base de datos se utiliza la Clasificación asociativa que es un campo integrado de la Minería de Reglas de Asociación (ARM) y Clasificación. El ARM tradicional fue diseñado teniendo en cuenta que los elementos tienen la misma importancia y en la base de datos simplemente se menciona su presencia o ausencia [18].

El presente artículo está estructurado de la siguiente forma: en la sección 2: Trabajos Previos, donde se verá las investigaciones realizadas anteriormente. Posteriormente en la sección 3, se mostrará los Métodos y Herramientas utilizados en el modelo difuso. En la Sección 4 se mostrará los Experimentos y Resultados que se obtuvieron luego, además de la comparación con otros artículos que analizaron la misma problemática. Finalmente, en la sección 5 Conclusiones y Trabajo a Futuro, se establecerá cuál fue el resultado obtenido mediante el modelo propuesto.

2. Trabajos relacionados

En el 2008 se propuso un nuevo índice de medida, como una alternativa en la toma de decisiones en cirugía bariátrica con respecto a su grado de obesidad, tomando como referencia el IMC y el porcentaje de grasa corporal en personas adultas, tomando como referencia los lineamientos de la OMS y la experticia de los autores para la generación de reglas [1]. En el contexto mundial de la salud el índice de masa corporal (IMC) es considerado el principal criterio para determinar si una persona tiene problemas de obesidad., sin embargo, el exceso de grasa en relación al porcentaje de grasa corporal (% BF), es en realidad el principal factor perjudicial en la enfermedad de la obesidad que por lo general se pasa por alto. El estudio propone un modelo difuso mediante la asociación de IMC y %BF.

En el 2013 se presenta un modelo difuso para la evaluación de la obesidad abdominal (AO) y el riesgo Cardio-Metabólico (CMR) [2]. Sobre la base de los indicadores más conocidos de mediciones antropométricas; como el índice de masa corporal (IMC), así como la circunferencia de la cintura (CC) y la relación cintura-altura (RCEst); se hizo la construcción de una función de pertenencia (OF), así también se mapeo los conjuntos difusos de IMC, AO.

En el 2014 se realizó el sistema de toma de diagnóstico médico difusos [3] para ayudar y apoyar a evaluar los pacientes con dolor torácico de acuerdo al grado de obesidad. Para este artículo se fuzzificó el índice de masa corporal (IMC). Ese mismo año [4] realizó un nuevo árbol de decisión difusa. Se propuso un algoritmo ID3 fuzzy generalizado en el uso generalizado de entropía de la información (abreviado como GFID3), que mejoró las deficiencias de algoritmos de árboles de decisión fuzzy existentes, ya que no toman en cuenta el impacto de la no linealidad; por lo cual son incapaces de integrar la selección de los atributos extendidos. En el 2015 se hizo el estudio en los niños de 4 a 6 años [5], en el estudio se tienen en cuenta el índice de masa corporal y la actividad física que realizan para el análisis de la obesidad en los niños.

3. Métodos y herramientas

La fuente de datos que se utilizó son un conjunto de 5962 registros médicos de personas entre los 6-17 años de edad, proporcionadas por instituciones educativas de Brasil. Los parámetros seleccionados y que fueron empleados para la predicción se muestran en la Tabla 1.

Posteriormente, se procedió al calculó del IMC de acuerdo a las pautas clínicas dada por [6], de acuerdo al análisis realizado sobre los datos recogidos, este cálculo se realizó

con respecto a los percentiles de edad, altura y peso y sexo. Después de calcularse el IMC en los niños y adolescentes, los valores percentiles del IMC y edad se muestran en la figura 1, proporcionada por los Centros para el Control y la Prevención de Enfermedades (CDC) para el IMC por edad para niñas para obtener la categoría del percentil. El IMC por edad para niños y niñas para obtener la categoría del percentil se muestran en la Fig. 1 y 2.

Tabla 1. Parámetros de entrada.

Parámetros	**Descripción**
Percentil Estatura	Índice de clasificación de la estatura por edad basado en percentiles (0-100).
Percentil Peso	Índice de clasificación del peso por edad basado en percentiles (0-100).
BF	Porcentaje de grasa

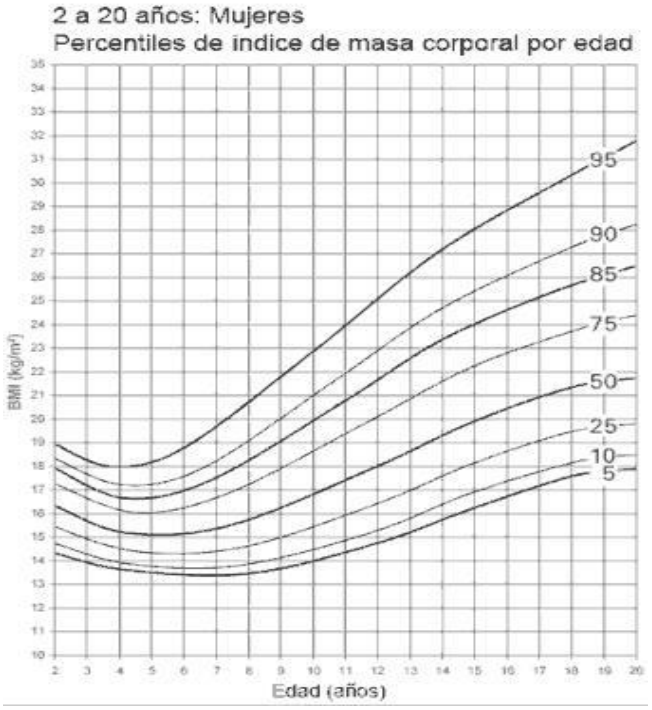


Fig.1. Índice de masa corporal en niñas basado en percentiles [6].

Los percentiles son el indicador que se utiliza con más frecuencia para evaluar el tamaño. El percentil indica la posición relativa del número del IMC del niño entre niños del mismo sexo y edad. Las categorías del nivel de peso del IMC por edad y sus percentiles correspondientes se muestran en la siguiente Tabla 1.

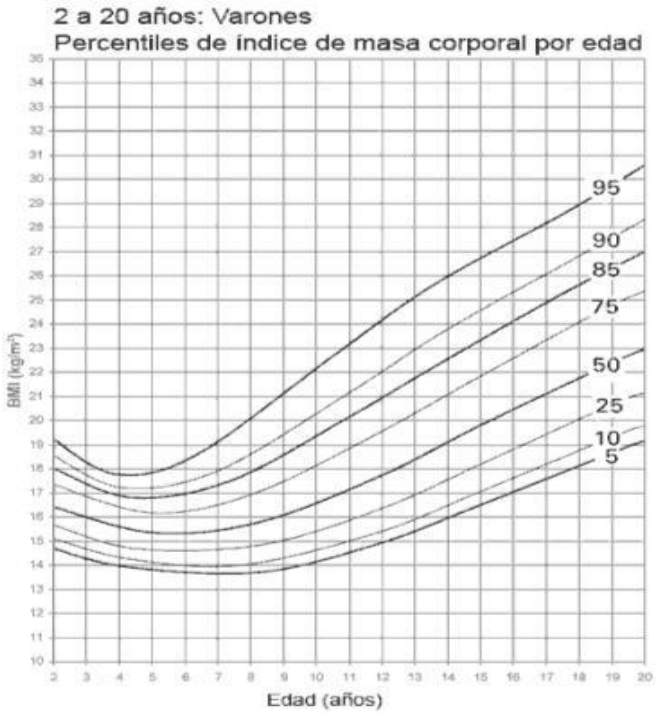


Fig.2. Índice de masa corporal en niños basada en percentiles [6].

Tabla 2. Pautas clínicas en la identificación de la obesidad en los niños [6].

Clasificación	Percentil IMC
Bajo peso	<5
Normal	>=5 y <85
Sobrepeso	>=85 y <95
Obesidad	>=95

Tabla 3. Pautas clínicas en la valoración del porcentaje de grasa en los niños [7].

Clasificación	Porcentaje de grasa
Muy Bajo	>0 y <5.5
Bajo	>=5.5 y <11.5
Rango Optimo	>=11.5 y <21
Moderadamente Alto	>=21 y <26
Alto	>=26 y<31
Muy Alto	>=31 y <=100

El %BF se considera un mejor índice para evaluar la obesidad que el IMC ya que este último sólo funciona como un indicativo del exceso de peso y el primero es un indicativo de exceso de la masa grasa y el riesgo de seguir siendo obesos [1][7]. Por lo tanto, %BF ha sido suficientemente aprovechado en este artículo. Los valores para niñas y niños se muestran en las Tablas 2 y 3.

Tabla 4. Pautas clínicas en la valoración del porcentaje de grasa las niñas [7].

Clasificación	Porcentaje de grasa
Muy Bajo	>0 y <11
Bajo	>=11 y <15
Rango Optimo	>=15 y <23.5
Moderadamente Alto	>=23.5 y <31.5
Alto	>=31.5 y<34.5
Muy Alto	>=31 y <=100

Para introducir un mecanismo coherente y hacer frente a la obesidad en este artículo se consideró tanto el percentil de peso, el percentil de estatura y %BF para presentar un nuevo índice de la obesidad en niños mediante la teoría de la lógica difusa.

4. Experimentos y resultados

Se consideró 4 etapas para desarrollar el trabajo, estos son: Pre-procesamiento de datos, Fuzificación, Generación de Reglas, Aplicación del Sistema Mandani, Clasificación de Resultados. Además de la comparación del Sistema Difuso con un clasificador usando un algoritmo *Machine Learning*, que en este caso es un árbol de decisión C4.5.

4.1 Pre-Procesamiento de datos

En la etapa inicial, se disponía de una base de datos que poseía un total de 14 atributos entre edad, sexo, índices de masa corporal, variables de esfuerzo físico, pliegues cutáneos, etc. De acuerdo a [1] se establece los valores Crisp para poder delimitar los límites y rangos de los atributos IMC y BF, sin embargo estos medidas son adecuadas para personas adultas, además se debe establecer de acuerdo a los expertos cuales son los atributos de medida que son factores críticos en la predicción de obesidad [7, 1]. En su trabajo de investigación [1] propone un nuevo índice de clasificación de obesidad para adultos tomando como referencia IMC y el porcentaje de grasa, tomando como referencia los resultados dados por [1] se tomó como entradas al sistema, el porcentaje de grasa corporal, así como los percentiles de peso y edad, ya que el modelo propuesto está orientado a personas de 6-17 años. Los valores de porcentaje de grasa (BF) tienen valores Crisp distintos para hombres y mujeres, por lo que fue necesaria una clasificación de los registros de datos en hombres y mujeres. Otro punto que se considero es la conversión de los atributos de BMI, peso y estatura a un percentil, debido a que como lo menciona la OMS [8] una medida básica para el control

de la obesidad es el percentil del BMI, ya que esto permite poder establecer un porcentaje y compararlo con otras evaluaciones hechas a otros niños y ver si está en un rango aceptable. Se tomó una plantilla base para la conversión de BMI a Percentiles de Luis F. Romero, investigador de la universidad de Málaga [9]. El siguiente paso fue una clasificación de los datos en hombres y mujeres así como la clasificación de sus registros médicos en cada una de las clases Crisp para el cálculo del promedio de cada uno de los atributos, así como la desviación estándar que serán utilizados en la etapa de fuzificación.

4.2 Fuzzificación

Una vez con los datos preparados se necesitaba tener los rangos y clases para cada uno de estos atributos cada uno de estos se basaron en una guía de expertos como es la OMS y otros artículos de investigación sobre Obesidad Infantil. El primero que es para el caso de Percentiles de BMI, peso y estatura se pudo conseguir gracias a la OMS en [10] que dispone de una serie de gráficas y tablas de percentiles tanto para niños como para niñas siendo las clases: bajo peso, peso normal, sobrepeso, y obesidad. En cuanto al Porcentaje de Grasa de acuerdo a un artículo visto en [7] donde se establecía de acuerdo a los pliegues cutáneos el resultado de la masa grasa, masa magra y el porcentaje de grasa corporal igualmente para niños y niñas siendo las clases: muy bajo, bajo, normal, moderado, alto y muy alto. Hay que considerar también la salida de clasificación, esto es importante porque se necesita datos pre-iniciales o de entrenamiento que servirá en la etapa de generación de reglas. Con los rangos y clases ya tomados fueron utilizados como entrada al sistema Mandani para lógica difusa.

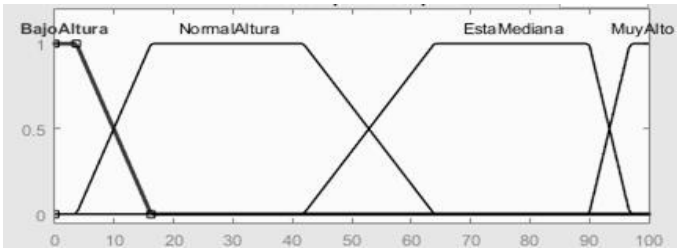


Fig.3. Índice Percentil Estatura en niños.

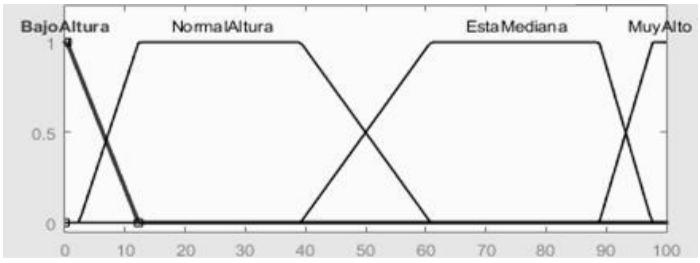


Fig.4. Índice Percentil Estatura en niñas.

En este caso se seleccionó la función de pertenencia y la que más se adapta al contexto del problema es la función de pertenencia de forma trapezoidal de acuerdo a

[1], estas funciones de pertenencia fueron establecidas mediante la función trapezoidal en base al promedio y las desviaciones estándares de cada rango para poder conseguir los cuatro parámetros de cada trapecio. La función de pertenencia trapezoidal se muestra en la ecuación 1, donde x es el atributo de entrada a ser fuzzificado, a, b, c, d son los rangos del trapecio.

$$f(x,a,b,c) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ 1, & b \le x \le c \\ \dfrac{d-x}{d-c}, & c \le x \le d \\ 0, & d \le x \end{cases} (1) \qquad (1)$$

Finalmente se obtuvo los resultados de cada atributo con sus valores fuzificados o graduados pudiendo pertenecer a más de una clase. En las siguientes Figuras 3, 4, 5, 6, 7, 8, se muestran las funciones de pertenencias de cada atributo de entrada.

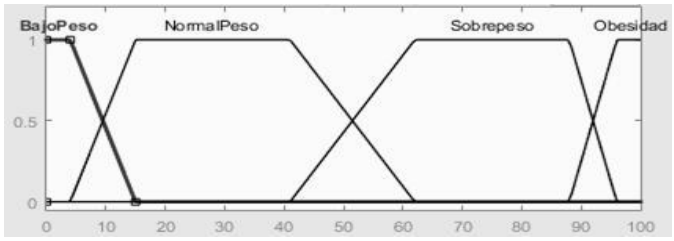


Fig.5. índice Percentil Peso en niños.

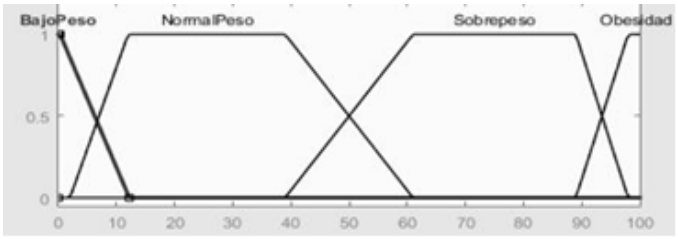


Fig.6. Índice Percentil Peso en niñas.

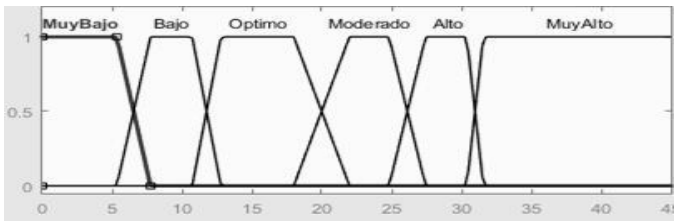


Fig.7. Índice BF en niños.

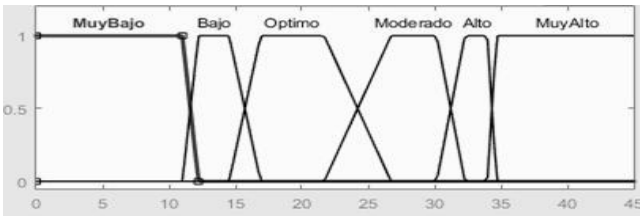


Fig.8. Índice BF en niña.

4.3 Generación de reglas

Para esta etapa, existen diversas herramientas con base en algoritmos de máquinas de aprendizaje que son generalizadas para trabajar con datos difusos, de acuerdo al estado del arte existen muchas opciones que depende mucho sobre los datos que se estén trabajando, así como la cantidad que representa, algunas técnicas poseen ventajas en cuanto a la robustez, pero también cuentan con desventajas como caer en óptimos locales. Otras técnicas permiten conseguir reglas muy óptimas, pero requieren de gran esfuerzo computacional para procesarlo. Y algunos carecen de métodos de simplificación, para sí obtener reglas con alta precisión, existen desde redes neuronales [11], mapas cognitivos [12], arboles de decisión [13, 14, 15] o algoritmos genéticos [16], todos generalizados para trabajar con datos fuzzificados.

Árbol ID3 Fuzzificado

El algoritmo ID3 fuzzificado, es una extensión del algoritmo clásico ID3, como un método para construir un árbol de decisión. En la construcción recursiva del árbol, se adopta una estrategia 'divide y vencerás'. El algoritmo Fuzzy ID3 calcula la entropía de partición difusa basada en el atributo y luego selecciona el atributo con la mínima partición de entropía difusa con una extensión del atributo, la entropía de partición es una extensión de la entropía de partición Crisp del algoritmo ID3 [17].

Sea $\Omega = \{1,2,\dots,n\}$, un conjunto de entrenamiento, $A = \{A_1 A_2,\dots,A_m\}$ un conjunto de atributos, donde $\{A_{i1}, A_{i2},\dots,A_{iki}\}$ es el rango del atributo A_i, en el árbol ID3 con valores fuzzificados, los nodos de decisión se consideran como conjuntos difusos en Ω., la entropía de partición difusa de un nodo no hoja D está dado por $FE(D, A_i)$ [4]:

$$FE(D, A_i) = \sum_{j=1}^{ki} \frac{m_{ij}}{mi} E\left(D, A_{ij}\right), \tag{2}$$

$$E\left(D, A_{ij}\right) = -\sum_{k=1}^{ni} \frac{m_{ijk}}{\overline{m_{ij}}} log_2 \frac{m_{ijk}}{\overline{m_{ij}}}, \tag{3}$$

donde $m_{ij} = M\left(A \cap A_{ij}\right), m_i = \sum_{j=1}^{k} m_{ij}$, $m_{ijk} = M\left(D \cap A_{ij} \cap C_k\right), \overline{m_{ij}} = \sum_{k=1}^{n} m_{ijk}$.

Una de las características del árbol ID3 difuso basado en la generalización de entropía (FID3) propuesto por [17], es que puede ser aplicado tanto a conjunto de datos con valores de atributos fuzzificados como a atributos numéricos y continuos.

El método elegido fue un árbol de decisión fuzzificado con el algoritmo ID3 generalizado para este propósito [17]. Aparte de considerar los buenos resultados que proporciona esta técnica, es necesario considerar sobre todo la rapidez con la cual puede proporcionar las reglas. Las reglas obtenidas, así como la tasa de precisión para cada clase de salida se muestran en las Tablas 4 y 5.

Tabla 5. Reglas difusas para Niñas.

Nro	Reglas	Precisión
1	IF Peso IS Normal AND Estatura IS MuyAlta AND BF IS MuyAlto THEN BajoPeso	1.00
2	IF Peso IS Sobrepeso AND BF IS Optimo AND Estatura IS Bajo THEN Obesidad	1.00
3	IF Peso IS Sobrepeso AND BF IS Alto AND Estatura IS Bajo THEN Sobrepeso	1.00
4	IF Peso IS Sobrepeso AND BF IS MuyAlto THEN Obesidad	1.00
5	IF Peso IS Normal AND Estatura IS Bajo THEN Sobrepeso	0.98
6	IF Peso IS Normal AND Estatura IS Normal AND BF IS MuyBajo THEN BajoPeso	0.90
7	IF Peso IS Sobrepeso AND BF IS Bajo AND Estatura IS Alta THEN Normal	0.90
8	IF Peso IS Normal AND Estatura IS MuyAta AND BFIS Moderado THEN BajoPeso	0.88
9	IF Peso IS Sobrepeso AND BF IS Bajo AND Estatura IS Normal THEN Normal	0.86
10	IF Peso IS Sobrepeso AND BF IS Moderado THEN Sobrepeso	0.84

Tabla 6. Reglas difusas para Niños.

Nro	Reglas	Precisión
1	IF Peso IS BajoPeso AND Estatura IS Normal AND BF IS Moderado THEN BajoPeso	1.00
2	IF Peso IS Normal AND Estatura IS Bajo AND BF IS Optimo THEN Normal	1.00
3	IF Peso IS Normal AND Estatura IS Bajo AND BF IS Alto THEN Sobrepeso	1.00
4	IF Peso IS Normal AND Estatura IS Normal AND BF IS Moderado THEN Sobrepeso	1.00
5	IF Peso IS Normal AND Estatura IS Normal AND BF IS Alto THEN Sobrepeso	1.00

Nro	Reglas	Precisión
6	IF Peso IS Sobrepeso AND BF IS MuyBajo AND Estatura IS Normal THEN Normal	1.00
7	IF Peso IS Sobrepeso AND BF IS MuyBajo AND Estatura IS SobreAlto THEN Normal	1.00
8	IF Peso IS Sobrepeso AND BF IS MuyBajo AND Estatura IS Alto THEN BajoPeso	1.00
9	IF Peso IS Sobrepeso AND BF IS MuyAlto AND Estatura IS Bajo THEN Obesidad	1.00
10	IF Peso IS Sobrepeso AND BF IS MuyAlto AND Estatura IS Alto THEN Sobrepeso	1.00

Con esto finalmente se pudo insertar dentro del requerimiento de reglas del sistema difuso para la clasificación de peso infantil, siendo dichas reglas las que pasaran a formar parte del resultado o salida del sistema difuso.

4.4 Clasificación y resultados

Después de proporcionar el conjunto de entrenamiento con un total de 5962 registros; 20% de los datos fueron utilizados para entrenar el árbol GFID3, mientras que el conjunto de datos total fueron empleados para test. El modelo difuso generado se muestra en la Fig. 9.

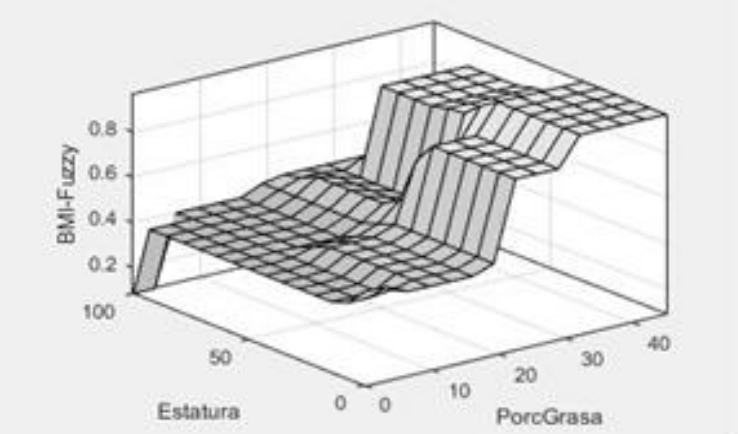


Fig.9. Relación Porcentaje de Grasa vs Estatura del Sistema de Clasificación Difusa para Niños.

En cuanto a los resultados de comparación se evaluó la categoría de precisión de las reglas, así como de las salidas. En la Tabla 6 y 7, se muestras algunos resultados de los primeros 10 registros para hombres y mujeres respectivamente.

Los resultados de clasificación en 2,935 niños y 3,021 en niñas estos resultados se muestran en porcentajes en las Tablas 8 y 9.

Tabla 7. Comparación de Salidas para Niños.

Estatura Percentil	Peso Percentil	%Grasa	Salida	Clasificación del Modelo	Salida Experto	Clasificación Experto
56.56	37.03	11.9	38.3	Normal	27.273	Normal
83.45	54.57	10.1	43.40	Normal	24.03	Normal
52.75	55.81	11.5	43.85	Normal	60.84	Normal
86.76	96.28	27.8	97.54	Obesidad	96.21	Obesidad
99.93	99.97	33.5	97.54	Obesidad	99.71	Obesidad
97.37	99.41	28.6	97.54	Obesidad	98.70	Obesidad
87.30	99.28	30.2	97.541	Obesidad	99.13	Obesidad
98.58	97.00	18.5	85.19	Sobrepeso	91.24	Sobrepeso
96.40	96.66	17.8	85.31	Sobrepeso	93.15	Sobrepeso
97.98	97.14	16.7	85.43	Sobrepeso	92.35	Sobrepeso

Tabla 8. Comparación de Salidas para Niñas.

Estatura Percentil	Peso Percentil	%Grasa	Salida	Clasificación del Modelo	Salida Experto	Clasificación Experto
82.40	30.15	13.7	7.38	Normal	1.69	Bajo Peso
78.08	58.84	11.9	45.29	Normal	37.25	Normal
84.90	54.29	15.4	43.82	Normal	19.87	Normal
100.00.	100.00	36.2	97.53	Obesidad	99.82	Obesidad
86.53	99.28	30.7	97.53	Obesidad	99.38	Obesidad
98.73	99.60	33.8	97.53	Obesidad	98.92	Obesidad
98.39	99.57	24.6	97.53	Obesidad	98.92	Obesidad
30.76	73.25	25.9	85.56	Sobrepeso	91.87	Sobrepeso
96.62	96.80	29.7	91.88	Sobrepeso	93.32	Sobrepeso
70.34	90.57	30.7	86.08	Sobrepeso	94.02	Sobrepeso

Tabla 9. Resultados de Clasificación para Niños.

Niños	Precisión
Correctamente Clasificados	83.65
Incorrectamente Clasificados	16.35

Tabla 10. Resultados de Clasificación para Niñas.

Niñas	Precisión
Correctamente Clasificados	76.13
Incorrectamente Clasificados	23.87

4.5 Discusiones

De acuerdo al sistema difuso y los resultados de clasificación con el algoritmo difuso basado en árboles de decisión GFID3, se observa claramente que el sistema difuso es mejor en cuanto a la clasificación dando un promedio de precisión de reglas 82.79% para niños y un porcentaje de 76.37% de precisión para niñas. A diferencia del estudio realizado por Miyahira y su índice difuso basado en IMC y BF propuesto para la predicción de casos de obesidad en personas adultas, el presente estudio tomo como entrada al sistema difuso los valores percentiles de IMC, peso, estatura y %BF, considerando la distribución de estos en el rango de edad de 2-20 años y generando las reglas más optimas mediante un conjunto de 5643 registros médicos y la aplicación del GFID3 generalizado que ha demostrado ser más robusto que las variantes propuestas del árbol ID3 y por tanto adecuado en el campo médico.

5. Conclusiones y trabajo a futuro

Este trabajo presenta un índice difuso para la predicción de casos de obesidad tanto en hombres como mujeres, tomando como entradas los Percentiles de peso, estatura y el %BF, basado en la lógica y los conjuntos difusos. Los resultados obtenidos son un conjunto de reglas y un modelo difuso que permite determinar si una persona es propensa o no a problemas de obesidad partir de sus registros médicos de peso, estatura y %BF, estas reglas fueron generadas mediante la fuzzificación de las entradas y como resultado del árbol GFID3 que ha demostrado tener gran exactitud frente a las variantes propuestas del ID3. El índice difuso desarrollado puede ser utilizado en el campo médico para la predicción de casos de obesidad, sin embargo, no está destinado a reemplazar o sustituir las métricas de evaluación empleados por médicos con experiencia; por el contrario, se debe ver como una herramienta de apoyo para la predicción de casos de obesidad en personas de 6-17 años. Como trabajos futuros se pretende extender este estudio para personas adultas, optimizando el conjunto de reglas generadas mediante un algoritmo evolutivo como los algoritmos genéticos y probar y validar sus resultados en el campo médico.

Referencias

1. Miyahira, S.A., de Azevedo, J.L.M.C., Araújo, E.: Fuzzy obesity index (MAFO I) for obesity evaluation and bariatric surgery indication. J. Transl. Med., Vol. 9, No. 1, p. 134 (2011)
2. Nawarycz, T., Pytel, K., Drygas, W., Gazicki-Lipman, M., Ostrowska-Nawarycz, L.: A fuzzy logic approach to the evaluation of health risks associated with obesity. Computer Science and Information Systems (FedCSIS) (2013)
3. Orsi, T., Araujo, E., Simoes, R.: Fuzzy chest pain assessment for unstable angina based on Braunwald symptomatic and obesity clinical conditions. Fuzzy Systems (FUZZ-IEEE), IEEE International Conference on, pp. 1076–1082 (2014)
4. Jin, C., Li, F., Li, Y.: A generalized fuzzy ID3 algorithm using generalized information entropy. Knowledge-Based Syst., Vol. 64, pp. 13–21 (2014)
5. Khanna, M., Srinath, D.N.K., Mendiratta, D.J.K.: The Study Of Obesity In Children Using Fuzzy Logic. IJITR, Vol. 3, No. 1 (2015)

6. OMS, (2007-2009). Comisión para acabar con la obesidad infantil: Datos y cifras sobre obesidad infantil Publishing. Recuperado de http://www.who.int/end-childhood-obesity/facts/es/.
7. Hall López, J.A., Monreal Ortiz, L.R., Ochoa Martínez, P.Y. Alarcón Mesa, E.: Porcentaje de grasa corporal en niños de edad escolar. Vol. 2, No. 1, pp. 13–17 (1988)
8. OMS, Centro para el Control y Prevención de Enfermedades CDC, Disponible en: http://www.cdc.gov/healthyweight/spanish/assessing/bmi/childrens_bmi/acerca_indice_ma sa_corporal_ninos_adolescentes.html (2015)
9. Romero, L.F.: Tabla de Crecimiento (Control de Percentiles), Disponible en: http://www.ac.uma.es/~felipe/personal.html (2009)
10. OMS, CDC growth charts to monitor growth for children age 2 years and older in the U.S. Disponible en: http://www.cdc.gov/growthcharts/clinical_charts.htm
11. Jang, J.-S.R.: ANFIS adaptive-network-based fuzzy inference system. Systems, Man and Cybernetics, IEEE Transactions on, Vol. 23, No.3, pp.665–685 (1993)
12. Kosko, B.: Fuzzy cognitive maps. Int. J. Man. Mach. Stud., Vol. 24, No. 1, pp. 65–75, Jan. (1986)
13. Lee, K.M., Lee, K.M. Lee, J.H., Lee-Kwang, H.: A fuzzy decision tree induction method for fuzzy data. Fuzzy Systems Conference Proceedings (1999)
14. Yuan, Y., Shaw, M.J.: Induction of fuzzy decision trees. Fuzzy Sets Syst., Vol. 69, No.2, pp. 125–139 (1995)
15. Janikow, C.Z.: Fuzzy decision trees: issues and methods. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, Vol.28, No.1, pp. 1–14 (1998)
16. Yuan, Y., Zhuang, H.: A genetic algorithm for generating fuzzy classification rules. Fuzzy Sets Syst., Vol. 84, No. 1, pp. 1–19 (1996)
17. Jin, C., Li, F., Li, Y.: A generalized fuzzy ID3 algorithm using generalized information entropy. Knowledge-Based Syst., Vol. 64, pp. 13–21 (2014)
18. Soni, S., Vyas, O.P.: Fuzzy Weighted Associative Classifier: A Predictive Technique for Health Care DataMining. Int. J. Comput. Sci. Eng. Inf. Technol., Vol. 2, No. 1, pp. 11–22 (2012)
19. Bouharati, S., Bounechada, M., Djoudi, A., Harzallah, D., Alleg, F., Benamrani, H.: Prevention of Obesity using Artificial Intelligence Techniques. International Journal of Science and Engineering Investigations, Vol. 1, No. 9, pp. 146–150 (2012)

Predicción de caudales medios diarios en la cuenca del Amazonas aplicando redes neuronales artificiales y el modelo neurodifuso ANFIS

Walter Enrique Béjar Chacón, Kid Yonatan ValerianoValdez,
Julio Cesar Ilachoque Umasi, Jose Sulla Torres

Universidad Nacional San Agustín, Escuela Profesional de Ingeniería de Sistemas, Arequipa, Perú

{wbejarch, jim.kidv, jcilachoque, josullato }@gmail.com

Resumen. El presente artículo muestra los resultados obtenidos al aplicar redes neuronales y el modelo neurodifuso ANFIS para la predicción de caudales medios diarios en una sección de la cuenca del Amazonas. Se aplicaron los pasos que componen la metodología KDD - Knowledge Discovery in Databases sobre la información hidrológica recolectada de las estaciones del servicio de observación SO HYBAM y sobre la información climatológica recolectada del Instituto de Investigación Internacional de Clima y Sociedad. La simulación se llevó a cabo en el software Matlab® donde se evaluó el comportamiento de ambos modelos, el estudio mostró que al utilizar técnicas de Inteligencia Artificial se consiguió coeficientes de correlación - CC superiores al 97%, un error medio porcentual absoluto - MAPE por debajo del 10% y otras 4 métricas que fueron utilizadas en esta investigación. Los resultados muestran que estos modelos pueden etiquetarse como buenos modelos para la predicción, gracias a su capacidad predictiva en comparación con métodos tradicionales del tipo lineal.

Palabras clave: Predicción de caudales, redes neuronales artificiales, sistemas neurodifusos, ANFIS, inteligencia artificial.

Forecast of Average Daily Flows in the Amazon Basin Using Artificial Neural Networks and the Adaptive Neuro-fuzzy Inference System (ANFIS)

Abtract. This article shows the results obtained by applying neural networks and neurofuzzy model ANFIS for predicting average daily flows in a section of the Amazon basin. The steps in the KDD methodology (Knowledge Discovery in Databases) were applied on hydrological data collected from observation stations of the SO HYBAM and climatological information collected from the International Research Institute for Climate and Society. The simulation was performed in the software Matlab® where the behavior of both models were evaluated, the study showed that using artificial intelligence techniques achieved correlation coefficients - CC above 97%, a mean absolute percentage error -

MAPE by below 10% and 4 other metrics that were used in this research. The results show that these models can be labeled as good models for prediction, thanks to its predictive ability compared to traditional methods of linear type.

Keywords: Forecast prediction, artificial neural networks, fuzzy neuro systems, ANFIS, Artificial Intelligence.

1. Introducción

Diversos trabajos de investigación tales como [1, 2] muestran que la cuenca Amazónica ha sufrido durante las dos últimas décadas de estiajes fuertes pero también de grandes inundaciones (1999, 2009, 2012, 2014), cada uno de estos eventos fue catastrófico para las poblaciones ribereñas de los grandes ríos Amazónicos, en el Perú los niveles alcanzados en el año 2015 por el Río Amazonas llegaron a equivalentes de crecidas históricas.

Lo que lleva consigo a aplicar métodos de pronóstico hidrológicos generalmente usando regresiones lineales, dichos modelos miden la relación entre variables dependientes e independientes del fenómeno y utilizan los caudales como datos de entrada [3, 4]. Pero debido a la no linealidad de estos fenómenos y que adicionalmente para que estos modelos sean más precisos se necesita incluir otras variables del tipo hidrológico y morfológico, lo que hace que estos modelos no sean apropiados [5] y sea necesario utilizar otro tipo de modelos.

Actualmente, se han desarrollado diversos estudios de modelos de predicción que integran técnicas de Inteligencia Artificial, cuya estructura matemática flexible es capaz de identificar relaciones complejas no lineales entre los datos de entrada y salida, y en problemas en donde es difícil describir el proceso utilizando ecuaciones físicas [6]. Una de las técnicas más usadas en este campo son las redes neuronales artificiales - ANN que simulan el funcionamiento del cerebro para la resolución de problemas, otras técnicas de Soft Computing utilizadas son la lógica difusa, que permite tratar información imprecisa y también el modelo neurodifuso ANFIS, que es una combinación de la lógica difusa y las redes neuronales [7, 8].

Recientes estudios han desarrollado distintos modelos implementando técnicas de Inteligencia Artificial, como la realizada en Grecia en el Observatorio Nacional de Atenas (NOA), donde se pronosticó la precipitación máxima diaria aplicando modelos de redes neuronales artificiales, utilizando datos durante los años de 1891-2009. Los resultados de la investigación sugieren que dependiendo de la frecuencia del entrenamiento puede tener un impacto en la formación óptima de la predicción [9].
Partiendo de este panorama se presenta esta investigación desarrollada en una sección de la cuenca del Amazonas, cuyo objetivo es el pronóstico de caudales medios diarios aplicando redes neuronales artificiales y el modelo neurodifuso ANFIS, con los cuales se podría realizar estudios para el pronóstico de inundaciones.

2. Aspectos metodológicos

El proceso que siguió esta investigación fue el conocido Knowledge Discovery in Databases - KDD, el cual pone un especial énfasis en la búsqueda de patrones

comprensibles que pueden ser interpretados como conocimiento útil o interesante. Se siguieron los pasos que componen el proceso KDD, que se pueden observar en la Fig. 1 y que formaron la base para la obtención de resultados en esta investigación.

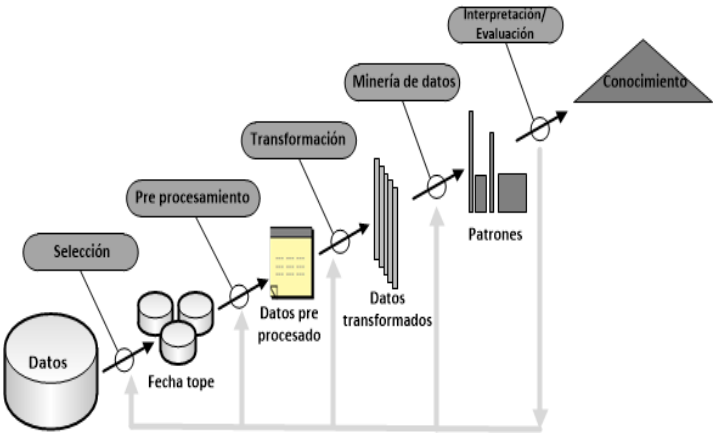


Fig. 1. Pasos que componen el proceso KDD [10].

2.1 Información hidrológica y climatológica

La cuenca hidrológica del Río Amazonas se ubica sobre terrenos de varios países de América del Sur: Perú, Colombia, Bolivia, Ecuador, Venezuela, Guyana, Surinam y Brasil, este último con la mayor extensión de cerca de 4 millones de km^2, que corresponden a casi la mitad de su superficie, la cuenca cubre una superficie de 6,2 millones de km^2. Así, la cuenca del Amazonas es la mayor cuenca hidrográfica en el mundo con un volumen de agua impresionante.

Se realizó una búsqueda de fuentes de datos hidrológicos libres, gracias al Servicio de Observación SO HYBAM (anteriormente Observatorio de Investigación del Medio ambiente) se pudo tener acceso a información de caudales de más de veintiuna estaciones que miden el caudal de la cuenca del Río Amazonas.

En cuanto a información climatológica, esta pudo ser obtenida gracias al Instituto Internacional de Investigación sobre el Clima y Sociedad, este posee un gran conjunto de datos muchos de ellos relacionados a la climatología, para esta investigación se utilizó el conjunto de datos brindado por la NOAA - National Oceanic and Atmospheric Administration, conjunto de datos que entre los atributos que posee se encuentra el atributo de precipitación media diaria [11].

2.2 Selección de datos

Para el presente estudio se escogió los datos de caudales medios diarios de 4 estaciones, las cuales son: Obidos, Fazenda Vista Alegre, Manacapuru y Serrinha, cuya localización se puede observar en la Fig. 2.

Dichas estaciones fueron escogidas ya que presentaban el menor número de datos faltantes respecto a los intervalos de tiempo en común.

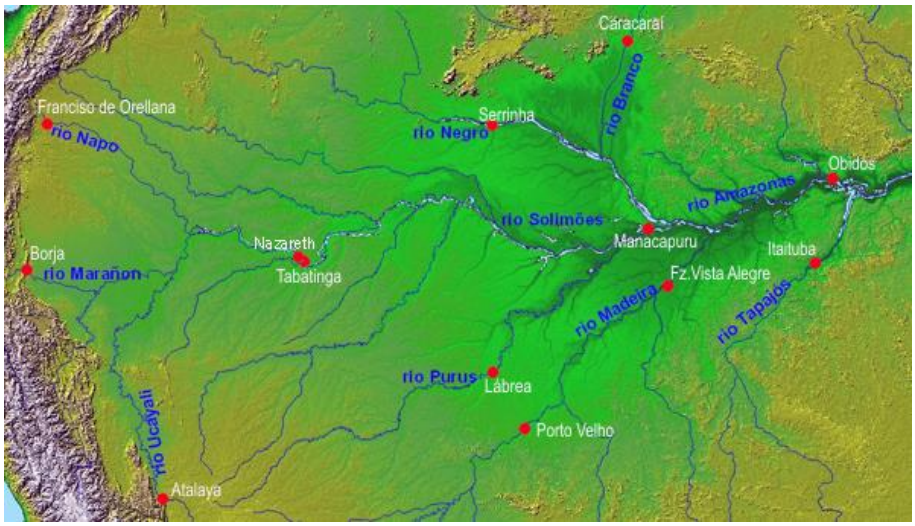


Fig. 2. Estaciones SO HYBAM ubicadas en la Cuenca del Amazonas.

En cuanto a los datos de precipitación solo se tomó en cuenta los datos obtenidos de una estación que está ubicada muy cerca de la desembocadura del Río Branco, dichos datos están comprendidos desde Marzo de 1989 a Diciembre de 1999, por lo que los datos de caudal fueron restringidos a dichas fechas.

En la Fig. 3 podemos ver los datos hidrológicos de la estación Obidos, dichos datos serán usados como salida de los modelos antes mencionados, ya que la porción del río donde se ubica esta estación es alimentada por el caudal de las otras 3 estaciones.

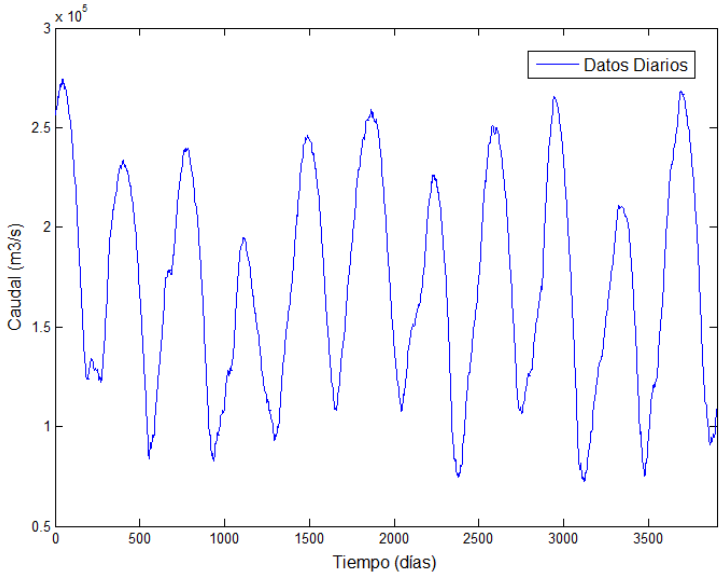


Fig. 3. Hidrograma estación Obidos 1989-1999.

2.3 Preprocesamiento

Como se pudo apreciar en la Fig. 3 los datos no presentan ruidos por lo que no es necesario realizar ninguna técnica de suavizamiento para los datos.

El inconveniente que se presentó en dicha etapa fue que había intervalos de tiempo en los cuales no se había registrado el caudal medio diario, esto se pudo dar por diferentes razones, como falla en los equipos de medición, falla en los registros de datos, etc. La cantidad de registros faltantes no sobrepasaba el 5% de la cantidad total de registros, por lo que los datos todavía podían ser utilizados [12], ya que la cantidad de registros faltantes no era significativa se procedió a estimar dichos valores en base a periodos de tiempo similares.

2.4 Transformación

Para esta etapa se optó por normalizar los datos, que consiste en re-escalar los valores de los datos dentro de un rango específico tal como -1 a 1 o de 0 a 1 [12][13].

Las ventajas de realizar una normalización a los datos hace posible acelerar la etapa de aprendizaje y evitar problemas numéricos tales como pérdida de precisión y desbordamiento aritméticos (overflows).

$$V' = 1/(1 + e^{-a}) \quad (1)$$

$$a = (V - mean)/std \quad (2)$$

Para los datos seleccionados se decidió realizar la normalización SoftMax que nos devuelve un rango de salida de 0 a 1. La normalización SoftMax utiliza las ecuaciones (1) y (2) para normalizar los datos de entrada, donde *V*: dato a normalizar, *mean*: media de los datos a normalizar, *std*: desviación estándar de los datos a normalizar y *V'*: dato normalizado.

2.5 Modelos de predicción

A continuación se describen las diferentes herramientas de ajuste y las metodologías de predicción que se usaron en esta investigación.

2.5.1 Redes neuronales artificiales

Las redes neuronales artificiales son modelos matemáticos inspirados en procesos neurobiológicos en los que el análisis de la información se imita a las acciones desarrolladas por las neuronas en el cerebro. La estructura estándar de una red neuronal artificial está compuesta por un conjunto de neuronas organizadas en capas (entrada, ocultas y de salida), distribuidas jerárquicamente; constituyendo un sistema funcional autónomo.

En este sistema inteligente, como el mostrado en la Fig. 4; se identifican los siguientes elementos: variables de entrada y salida, pesos sinápticos (que son la intensidad de interacción entre las neuronas), función de propagación, función de activación y función de salida. Además, se debe tener en cuenta el número de capas y

neuronas, ya que estos son los parámetros más importantes en la red neuronal artificial determinando así la eficiencia del sistema.

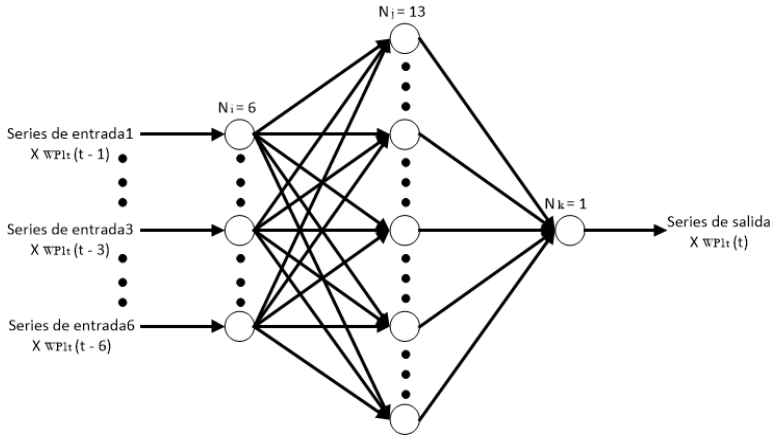


Fig. 4. Red Neuronal de 3 Capas [5].

Una de las ventajas de las redes neuronales artificiales es que pueden ser una herramienta útil para el modelado, cuando la relación entre los datos de entrada y salida no es clara o en su defecto es desconocida, razón por la cual este tipo de modelos son llamados de caja negra; lo que nos permite que a través de sus composiciones matemáticas sean capaces de modelar sistemas complejos como los sistemas hidrológicos. Otro de los beneficios derivados de las redes neuronales artificiales es su capacidad de generar salidas de una combinación específica de entradas y su capacidad de respuesta frente al manejo de datos no lineales, por lo que estos sistemas inteligentes pueden llegar a ser mejores que los sistemas de modelos lineales debido a su flexibilidad al abordar problemas complejos.

La simulación de las redes neuronales artificiales se realizó en el toolbox del software Matlab 2013® llamado nntool (neural network tool). La estructura de la red neuronal se compone de 4 conjuntos de datos de entrada (3 conjuntos de datos de caudal y un conjunto de datos de precipitación) y un conjunto de datos de salida. En cuanto a los parámetros utilizados para la calibración del modelo se estableció un tipo de entrenamiento backpropagation [14], debido a que la arquitectura implementada fue multi-capa, con neuronas ocultas, se utilizó la función de aprendizaje Levenberg-Marquardt (*trainlm*) que realiza mejor la función de ajuste para el reconocimiento de patrones del sistema y la función de error medio cuadrático (MSE).

Se utilizó un máximo de mil repeticiones para correr el modelo hasta llegar a la validación total del procesamiento de la información en conjunto con un gradiente mínimo de 1*e* - 07 y un máximo de seis revisiones de validación para evaluar la calidad del modelo. El modelo computacional divide los datos en tres muestras, el 70% del total de los datos ingresados sirven para el entrenamiento, 15% de los datos para la validación y el otro 15% para las pruebas del modelo.

Se procedió a simular distintos escenarios en los cuales se modificaba el número de capas (entre 2 y 5) y el número de neuronas (entre 3 y 10), generando así treinta y dos escenarios en los cuales se utilizó la función de propagación *sigmoidal - sigmoidal*.

En cada escenario entrenado se realizó la simulación del pronóstico durante el rango de fechas elegido y con los resultados obtenidos se hallaron cada una de las métricas de comparación por medio de un programa realizado en Matlab 2013®.

2.5.2 Modelo neurodifuso ANFIS

Para explicar que es un sistema neuro-difuso se debe partir del enfoque de la lógica difusa que se basa en expresiones lingüísticas inciertas en lugar de la incertidumbre numérica. Esta técnica se basa en un sistema de inferencia difusa (FIS) basado en tres componentes: una base de normas (rulebase) que contiene las reglas difusas si-entonces, una base de datos (database) definida por una función de pertenencia y un sistema de inferencia (interference system) que combina las reglas difusas y produce los resultados del sistema [15], la estructura de dicho sistema se muestra en la Fig. 5.

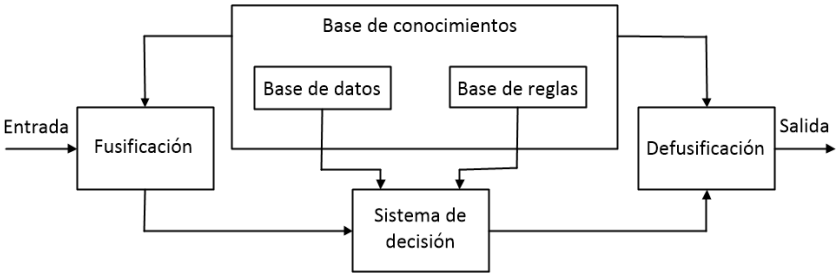


Fig. 5. Estructura Modelo de Inferencia Difuso [15].

La estructura de un sistema neuro-difuso consiste en la unión de dos técnicas artificiales inteligentes: redes neuronales artificiales y lógica difusa, razón por la cual es llamada ANFIS (Adaptive Neuro-Fuzzy Interference System). Este sistema utiliza los algoritmos de aprendizaje y la configuración por medio de capas y neuronas de las redes neuronales artificiales, mientras que de los sistemas de inferencia difusa utiliza el razonamiento difuso que permite generar reglas de inferencia a partir de la asignación de variables lingüísticas a la información caracterizando así la combinación de cada una de las entradas a una o varias salidas. Esta combinación permite a la red organizarse a sí misma y generar adaptabilidad del sistema difuso para resolver distintos problemas, en la Fig. 6 se muestra la estructura general del modelo ANFIS que se compone de 5 capas.

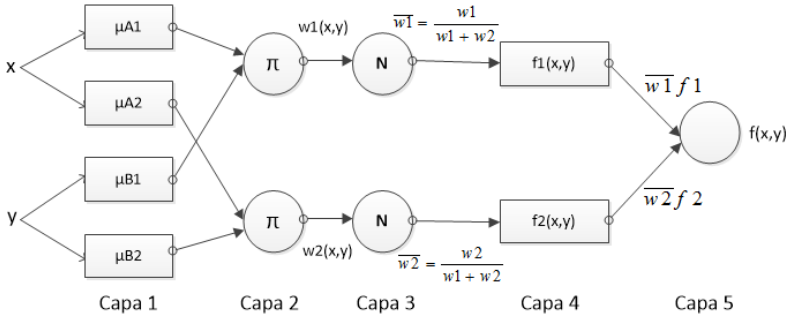


Fig. 6. Estructura Modelo ANFIS [16].

La simulación del modelo ANFIS se realizó en el toolbox del software Matlab 2013® llamado anfisedit. La estructura del modelo ANFIS se compone de 4 conjuntos de datos de entrada (3 conjuntos de datos de caudal y un conjunto de datos de precipitación) y un conjunto de datos de salida.

El conjunto de datos de entrada se dividió de la siguiente forma, el 70% del total de los datos ingresados sirven para el entrenamiento, 15% de los datos para la validación y el otro 15% para las pruebas del modelo. Se simularon en total 18 escenarios, variando la cantidad de épocas entre 10 y 15.

Respecto a las reglas de inferencia se usó un sistema tipo Sugeno, usando reglas de inferencia tipo producto (and) entre las entradas, funciones de pertenencia trapezoidal, campana y triangular junto a un método de defusificación tipo *wtaver* el cual retorna la media ponderada a la salida del sistema difuso del ANFIS.

3. Resultados

Para comparar la precisión y exactitud de los modelos inteligentes mostrados en esta investigación se usaron 6 datos estadísticos, usados en la mayoría de investigaciones observados y utilizados como métricas de evaluación de modelos simulados [5]. Estos son:

MAE (Mean Absolute Error): el error medio absoluto es una medida de precisión usada para evaluar pronósticos, en él se evalúa el valor absoluto promedio de la diferencia entre el dato real y el dato pronosticado, entre más pequeño sea el error o tienda a cero, será más preciso el pronóstico [17].

MSE (Mean Squared Error): el error medio cuadrático es una medida de precisión usada para evaluar pronósticos, en la que se evalúa el desempeño del valor promedio de la diferencia entre el dato real y el dato pronosticado al cuadrado [18].

MAPE (Mean Absolute Percentage Error): el error porcentual absoluto de la media permite analizar la exactitud del modelo en términos porcentuales, teniendo en cuenta el valor absoluto de la relación entre el dato real y el dato pronosticado. La escala para evaluar la exactitud del modelo usando el MAPE determina que un pronóstico muy exacto es el que tiene un valor menor o igual al 10%, un buen pronóstico tiene un valor entre un rango del 11% al 20%, un pronóstico razonable entre el 21% al 50% y un pronóstico inadecuado mayor al 50% [19].

RMSE (Root Mean Squared Error): la raíz cuadrada del error cuadrático medio es una medida de precisión usada para evaluar pronósticos, en ella se evalúa el valor de la raíz del promedio cuadrático de la diferencia entre el dato real y el dato pronosticado, entre más pequeño sea el error o tienda a cero, será catalogado como el pronóstico más preciso [17].

CC (Correlation Coefficient): el coeficiente de correlación de Pearson determina la relación del dato real y el dato pronosticado, en cuanto a la covarianza con las desviaciones de los dos tipos de datos, a diferencia del CCC, el coeficiente de correlación ignora componentes de exactitud [20].

CCC (Concordance Correlation Coefficient): el coeficiente de correlación de concordancia indica la relación entre la precisión y exactitud del modelo, este mide el grado en que la covarianza del dato real y el dato pronosticado se acercan a la recta de 45° del modelo, este factor se evalúa con valores entre 0 y 1 [21].

3.1 Resultados redes neuronales artificiales

Se eligieron 5 de los 32 escenarios con mayor CCC, dichos escenarios se compararon entre sí ponderando aquellas métricas que cumplían con la mayor cantidad de resultados favorables en los criterios estadísticos a evaluar con el MAE, MAPE, MSE, RMSE y CC para calcular la precisión y exactitud de los modelos. A continuación en la Tabla 1 se observan los 5 mejores escenarios.

Tabla 1. Mejores escenarios redes neuronales artificiales.

Escenario	CCC	CC	MAE	MAPE	MSE	RMSE
Esc 32	0.9937	0.9937	0.0173	4.5384	0.00061	0.0246
Esc 27	0.9934	0.9934	0.0182	4.8242	0.00064	0.0252
Esc 28	0.9932	0.9933	0.0182	5.0245	0.00065	0.0254
Esc 24	0.9930	0.9930	0.0183	4.8438	0.00067	0.0259
Esc 20	0.9920	0.9921	0.0198	5.1768	0.00076	0.0276

Luego de analizar todos los criterios en los 5 mejores escenarios, el mejor escenario es el número 32 debido a que cumple con la mayor cantidad de resultados favorables, menor MAE (0.0173), menor MAPE (4.5384%), menor MSE (0.00061), menor RMSE (0.0246), mayor CC (0.9937) y mayor CCC (0.9937). Tomando como referencia el resultado obtenido con el MAPE se puede decir que el pronóstico arrojado es exacto ya que se encuentra por debajo del 10%, el CC nos indica que el modelo presenta el 99% de precisión del pronóstico en cuanto a la relación de los datos reales con los datos simulados. En este escenario se implementaron 5 capas y 10 neuronas por cada capa, lo cual nos indica que al implementar una red neuronal artificial se debe evaluar el incrementar el número de capas y de neuronas para un pronóstico adecuado.

A continuación se observa la Fig. 7 donde se muestra la comparación entre los datos reales con el mejor escenario simulado. En este histograma se muestran los caudales medios diarios en el periodo comprendido desde Febrero de 1996 a Diciembre de 1999, que corresponde al 15% de datos de prueba. Se observa que el modelo basado en redes neuronales artificiales se acerca bastante a los datos de caudal real.

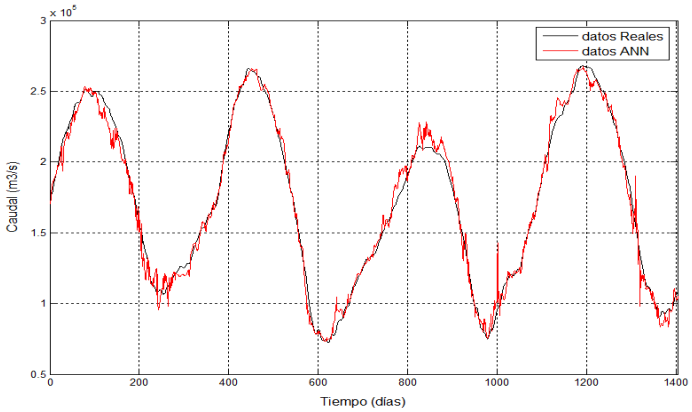


Fig. 7. Comparación Hidrogramas Reales vs ANN.

3.2 Resultados modelo neurodifuso ANFIS

De igual manera se escogieron los 5 mejores escenarios con el sistema neurodifuso, que son mostrados en la Tabla 2.

Tabla 2. Mejores escenarios modelo neurodifuso.

Escenario	CCC	CC	MAE	MAPE	MSE	RMSE
Esc 18	0.977	0.9772	0.0366	9.2452	0.0022	0.0465
Esc 17	0.9769	0.9772	0.0367	9.2607	0.0022	0.0466
Esc 16	0.9767	0.977	0.0368	9.3137	0.0022	0.0468
Esc 15	0.9763	0.9766	0.0371	9.3736	0.0022	0.0471
Esc 5	0.9762	0.9765	0.0373	9.3855	0.0022	0.0473

Luego de observador los resultados de la Tabla 3, se eligió el escenario 18 como el mejor pronóstico ya que cumple con la mayor cantidad de resultados favorables, menor MAE (0.0366), menor MAPE (9.2452%), menor MSE (0.0022), menor RMSE (0.0465), mayor CC (0.9772) y el mayor CCC (0.977). Este escenario fue simulado con una función de pertenencia de campana, con 15 épocas. La comparación con el mejor de los escenarios se muestra en la Fig. 8 donde figuran los caudales reales comprendidos en el periodo que va desde Febrero de 1996 a Diciembre de 1999, dichos datos corresponden al 15% de datos de prueba.

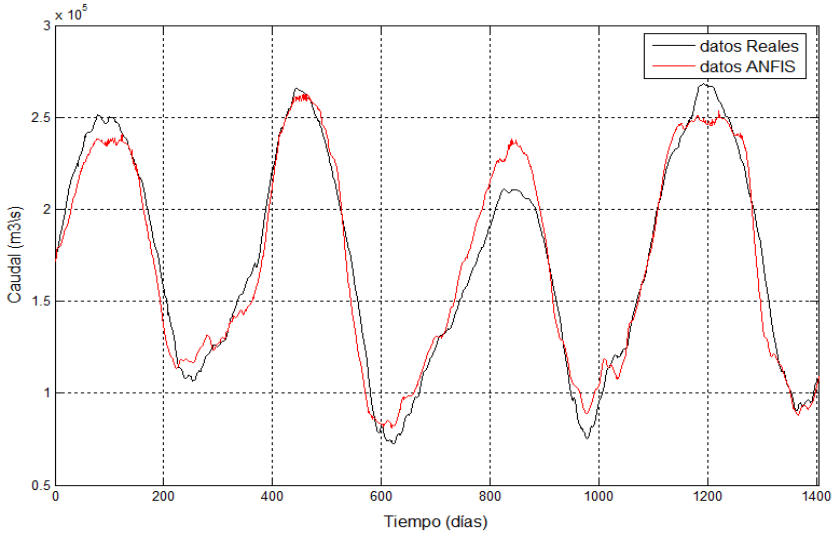


Fig. 8. Comparación Hidrogramas Reales vs ANFIS.

3.3 Resultados ANFIS vs red neuronal artificial

A continuación se muestra en la Tabla 3 los dos mejores modelos que se desarrollaron en esta investigación (red neuronal artificial sigmoidal - sigmoidal, modelo ANFIS con función de pertenencia en campana).

Tabla 3. Comparación del mejor escenario de los modelos planteados.

Modelo	CCC	CC	MAE	MAPE	MSE
Modelo ANN	0.9937	0.9937	0.0173	4.5384	0.00061
Modelo ANFIS	0.977	0.9772	0.0366	9.2452	0.0022

Los resultados obtenidos muestran una clara diferencia entre el modelo neurodifuso y el modelo basado en redes neuronales artificiales, siendo este último el que logra entregar los mejores pronósticos. Para observar mejor el comportamiento de los datos y el porqué de la elección del CCC, es necesario graficar los datos reales vs datos simulados comparándolos con la recta de la ecuación y=x.

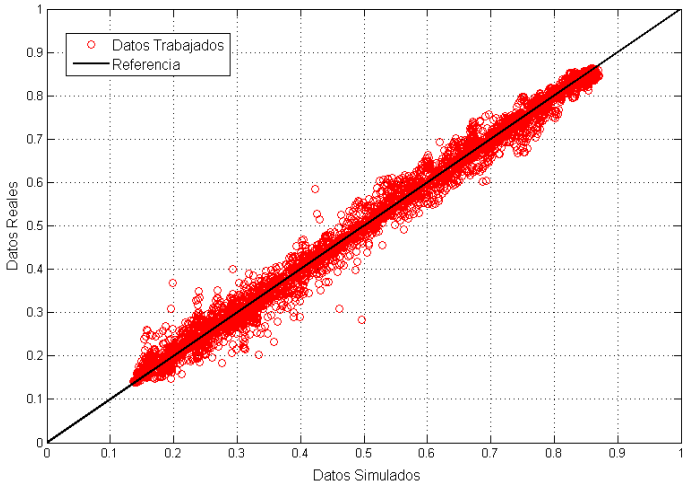


Fig. 9. Datos Reales vs Datos Simulados ANN.

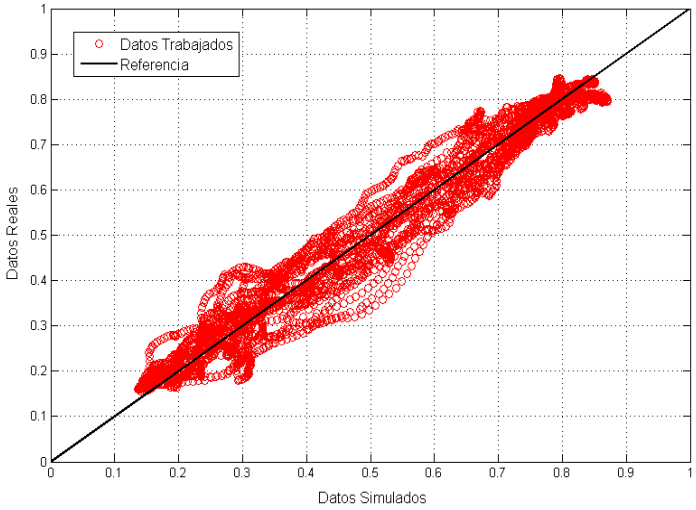


Fig. 10. Datos Reales vs Datos Simulados ANFIS.

En la Fig. 9 se hace la comparación de los datos reales y simulados por la red neuronal artificial donde podemos observar que los datos no están tan dispersos en comparación a la Fig. 10, en donde los datos se alejan de la recta de referencia. Gracias a las métricas utilizadas podemos observar que entre el dato pronosticado y el dato real el error es inferior al 10%.

4. Conclusiones

La implementación de modelos basados en técnicas de inteligencia artificial con los cuales se simulo el comportamiento de un tramo de la cuenca del Amazonas arrojó resultados estadísticos que demuestran pronósticos muy cercanos a los datos reales y la efectividad de dichas técnicas (MAPE entre 4% y 10% que indica un buen pronóstico; CC entre 0.97 y 0.99 que indica una buena relación entre los datos reales y los simulados; CCC entre 0.97 y 0.99 que indica demuestra precisión y exactitud en el pronóstico).

Se logró observar que las técnicas de inteligencia artificial (o de caja negra) como las redes neuronales artificiales arrojaron mejores resultados que los modelos basados en ANFIS, los modelos basados en redes neuronales artificiales pueden ser iguales o mejores que los modelos hidrológicos comunes basados en modelos no lineales y que la aplicación de este tipo de soluciones puede ser una buena alternativa para la generación de sistemas de alertas tempranas, integrando dicho sistema a sistemas de información geográfica que permitan visualizar la información pronosticada en estos sistemas.

Se recomienda continuar la investigación para la construcción de alertas tempranas de sistemas de información geográfica (SIG) y buscar otras zonas donde se pueda adaptar este proyecto para la preparación de la población ante riesgos de inundación.

Además, se recomienda la implementación del uso de técnicas artificiales inteligentes enfocadas al pronóstico de inundaciones en países de relieves tan diversos como los del continente Suramericano, es necesario continuar validando estos métodos no solo con los datos de 3 estaciones hidrológicas si no también integrando las medidas de estaciones cercanas o tener en cuenta otro tipo de variables tales como sociales, logísticas, económicas y culturales, lo que podría generar mayores desarrollos tecnológicos para no solo la creación de alertas tempranas sino también para la prevención de desastres naturales de distinto tipo.

Por otro lado surge la posibilidad de estudiar este problema con otro tipo de técnicas, lógica difusa intuitiva y algoritmos genéticos, comparando estos con modelos matemáticos y/o hidráulicos.

Referencias

1. Espinoza, J.C., Ronchail, J., Frappart, F., Lavado, W., Santini, W., Guyot, J.L.: The major floods in the amazonas river and tributaries (western amazon basin) during the 1970-2012 period: A focus on the 2012 flood*. Journal of Hydrometeorology, Vol. 14, No. 3, pp.1000–1008 (2013)
2. Marengo, J.A., Espinoza, J.C.: Extreme seasonal droughts and floods in amazonia: causes, trends and impacts. International Journal of Climatology (2015)

3. Pandey, G.R., Nguyen, V.T.V.: A comparative study of regression based methods in regional flood frequency analysis. Journal of Hydrology, Vol. 225, No. 1, pp. 92–101 (1999)
4. Weisberg, S.: Applied linear regression. John Wiley & Sons, Vol. 528 (2005)
5. Dawnson, C.W., Abrahart, R.J., Shamseldin, A.Y., Wilby, R.L.: Flood estimation at ungauged sites using artificial neural networks. Journal of Hydrology, Vol. 319, No. 1, pp. 391–409 (2006)
6. Seckin, N., Cobaner, M., Yurtal, R., Haktanir, T.: Comparison of artificial neural network methods with l-moments for estimating flood flow at ungauged sites: the case of east mediterranean river basin, turkey. Water resources management, Vol. 27, No. 7, pp. 2103–2124 (2013)
7. Kalteh, A.M.: Monthly river flow forecasting using artificial neural network and support vector regression models coupled with wavelet transform. Computers & Geosciences, Vol. 54, pp. 1–8 (2013)
8. Goyal, M.K., Bharti, B., Quilty, J., Adamowski, J., Pandey, A.: Modeling of daily pan evaporation in sub tropical climates using ann, ls-svr, fuzzy logic, and anfis. Expert Systems with Applications, Vol. 41, No.11, pp. 5267–5276 (2014)
9. Nastos, P.T., Paliatsos, A.G., Koukouletsos, K.V. Larissi, I.K., Moustris, K.P.: Artificial neural networks modeling for forecasting the maximum daily total precipitation at athens, greece. Atmospheric Research, Vol. 144, pp. 141–150, (2014)
10. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery in databases. AI magazine, Vol. 17, No. 3, p. 37 (1996)
11. Vose, R.S., Schmoyer, R.L., Peterson, T.C., Steurer, P.M., Heim Jr, R.R., Karl, T.R., Eischeid, J.K.: The global historical climatology network: Long-term monthly temperature, precipitation, and pressure data. Technical report, Oak Ridge National Lab., TN (United States) (1992)
12. Hernández, C., Rodríguez Rodríguez, J.E.: Preprocesamiento de datos estructurados. Vínculos, Vol. 4, No. 2, pp. 27–48 (2013)
13. Dapozo, G.N., Porcel, E., López, M.V., Bogado, V.S.: Técnicas de preprocesamiento para mejorar la calidad de los datos en un estudio de caracterización de ingresantes universitarios. IX Workshop de Investigadoresen Ciencias de la Computación (2007)
14. Kourentzes, N., Barrow, D.K., Crone, S.F.: Neural network ensemble operators for time series forecasting. Expert Systems with Applications, Vol. 41, No. 9, pp. 4235–4244 (2014)
15. Firat, M., Güngör, M.: River flow estimation using adaptive neuro fuzzy inference system. Mathematics and Computers in Simulation, Vol. 75, No. 3, pp. 87–96, (2007)
16. Firat, M.: Comparison of artificial intelligence techniques for river flow forecasting. Hydrology and Earth System Sciences, Vol. 12, No. 1, pp. 123–139 (2008)
17. Singhal, D., Swarup, K.S.: Electricity price forecasting using artificial neural networks. International Journal of Electrical Power & Energy Systems, Vol. 33, No. 3, pp. 550–555 (2011)
18. S. da S. Gomes, G., Ludermir, T.B.: Optimization of the weights and asymmetric activation function family of neural network for time series forecasting. Expert Systems with Applications, Vol. 40, No. 16, pp. 6438–6446 (2013)
19. Lewis, C.D.: Industrial and business forecasting methods: A practical guide to exponential smoothing and curve fitting. Butterworth Heinemann (1982)
20. Li, J., Feng, P., Wei, Z.: Incorporating the data of different watersheds to estimate the effects of land use change on flood peak and volume using multi-linear regression. Mitigation and Adaptation Strategies for Global Change, Vol. 18, No. 8, pp. 1183–1196 (2013)
21. Lin, L., Hedayat, A.S., Sinha, B., Yang, M.: Statistical methods in assessing agreement: Models, issues, and tools. Journal of the American Statistical Association, Vol. 97, No. 457, pp. 257–270 (2002)

Árboles de decisión ID3 para el diagnóstico de apendicitis aguda en niños

Iván Sánchez Martínez[1], Leticia Flores Pulido[1], Alfredo Adán Pimentel[2], J. Federico Ramírez Cruz[1], María del Rocío Ochoa Montiel[1]

[1] Universidad Autónoma de Tlaxcala, Facultad de Ciencias Básicas, Ingeniería y Tecnología,
Apizaco, Tlaxcala, México

[2] Universidad Autónoma de Tlaxcala, Facultad de Ciencias de la Salud, Zacatelco, Tlaxcala, México

{computoisz,aicitel.flores,ma.rocio.ochoa}@gmail.com,{aadanpimentel, federico_ramirez}@yahoo.com.mx

Resumen. La apendicitis aguda es una enfermedad de difícil diagnostico en ancianos y niños, en la mayoría de los casos se presenta un dolor en el ombligo que aumenta conforme al tiempo y se dirige hacia el cuadrante inferior derecho. Por su diversidad sintomatológica, existen propuestas de diagnóstico que se limitan a unos cuantos síntomas, signos y laboratorios. La apendicitis aguda por ser una de las enfermedades mortales y más comunes en el ser humano, ha sido abordada a nivel computacional para desarrollar modelos o conjuntos de reglas para diagnosticar de manera oportuna. En este trabajo se propone la creación de un árbol de decisión basado en arquitectura ID3 para la detección de apendicitis aguda en niños entre los 4 y 15 años en el Hospital General Regional "Lic. Emilio Sánchez Piedras"(Tzompantepec, Tlaxcala, México). Los datos para construir los árboles de decisión ID3 se recolectaron de manera prospectiva utilizando dos escalas de puntuación diagnostica (escala de Alvarado y puntuación de apendicitis pediátrica), incluyendo algunas variables demográficas y de confirmación de la enfermedad.
Se recolecto un total de 41 casos los cuales fueron analizados y base para la construcción de dos árboles de decisión ID3. Los resultados muestran un buen desempeño en el diagnóstico de apendicitis, sin embargo es necesario recolectar una mayor cantidad de datos para maximizar el grado de precisión y factibilidad.

Palabras clave: Apendicitis aguda, árbol de decisión ID3, diagnóstico computarizado, niños.

1. Introducción

La patología quirúrgica más frecuente en abdomen es la apendicitis. La apendicitis es actualmente el procedimiento quirúrgico de urgencia más común

en el mundo. Uno de cada 15-20 mexicanos presentará apendicitis aguda en algún momento de su vida. La apendicitis aguda representa la patología quirúrgica más común en la infancia y se presenta en 1-2 casos por cada 10,000 niños menores a 4 años y 25 casos por cada 10,000 en niños entre 4 y 17 años de edad. La sospecha y diagnóstico de apendicitis aguda se basa predominantemente en la clínica. El diagnóstico incorrecto o tardío aumenta el riesgo de complicaciones como infección de herida quirúrgica (8 - 15 %), perforación (5 - 40 %), abscesos (2 - 6 %), sepsis y muerte (0.5 - 5 %). El diagnóstico retardado incrementa costos en el servicio de urgencias y hospitalarios. Ésta enfermedad es muy común y sigue siendo un problema de diagnóstico y representa un reto para todos los médicos que atienden al paciente con sintomatología, a pesar de la experiencia y los diferentes métodos de diagnóstico clínico y paraclínico llegando a desconcertar hasta al mejor de los médicos [10].

Existen muchos intentos para mejorar el diagnóstico uno de ellos es el análisis en laboratorio, en cuanto a: el conteo de glóbulos blancos, neutrófilos y proteína C reactiva [7]. Más su valor diagnóstico es limitado. Otras técnicas más sofisticadas son las radiografías, ecografías y tomografías computarizadas, estas últimas tienen un alto riesgo de radiación ionizante innecesaria para el paciente [28], inaccesible en ciertas unidades médicas y de un costo elevado.

Existen pruebas menos invasivas que han demostrado buena precisión diagnostica como son los sistemas de puntuación diagnostico como la escala de Alvarado [1] y la puntuación de apendicitis pediátrica [23].

En la medicina es posible aplicar métodos alternativos de diagnóstico, debido a la gran cantidad de padecimientos involucrados, las sintomatologías y los pacientes. Lo ideal sería que los médicos pudieran contar con el apoyo de una herramienta que les permita analizar los datos sintomatológicos de cada uno de sus pacientes para poder determinar con base en casos anteriores, el diagnóstico más acertado, lo cual representaría un soporte y ayuda para el médico [3].

Una alternativa para la predicción y clasificación de datos, que es utilizada ampliamente en el área de la inteligencia artificial son los árboles de decisión. Diferentes estudios han demostrado que esta técnica es útil en el área médica principalmente en la toma de decisiones [3,27,26]. En el caso de la apendicitis otros autores han utilizado técnicas similares [13,18,25,20,28] aunque la mayoría de ellos, cae en el error de generalizar su muestra lo cual en el campo real sus modelos pueden ser ambiguos y no eficaces en ciertas categorías de la muestra, en este caso los niños.

En este artículo, se propone construir una herramienta de diagnóstico computacional utilizando arboles de decisión ID3 para el diagnóstico de apendicitis aguda en niños entre los 4 y 15 años.

Para la recolección de datos se hizo uso de encuestas en base las escalas de Alvarado y puntuación de apendicitis pediátrica con el fin de determinar que escala es la mejor opción para nutrir la base de conocimientos de un árbol de decisión ID3. El espacio de recolección fue el área de urgencias del Hospital General Regional "Lic. Emilio Sánchez Piedras" (Tzompantepec, Tlaxcala, México) donde se recolectaron un total de 41 casos.

La programación, implementación y prueba de los árboles de decisión ID3 se realizo en Python [21] y Weka (Waikato Environment for Knowledge Analysis, que en español se traduce como "Entorno para análisis del conocimiento de la Universidad de Waikato") [12].

Los resultados de este trabajo utilizando las escalas de evaluación y árboles de decisión son satisfactorios, sin embargo, es necesario un número mayor de casos para mejorar la exactitud diagnostica de nuestro proyecto.

2. Conceptos básicos

2.1. Apendicitis aguda

La infamación aguda del apéndice vermiforme, apendicitis, es una causa frecuente de abdomen agudo (dolor abdominal intenso de aparición súbita). La apendicitis es un proceso evolutivo, secuencial, de allí las diversas manifestaciones clínicas y anatomopatólogicas que suele encontrar el cirujano y que dependerán fundamentalmente del momento o fase de la enfermedad en que es abordado el paciente [14].

2.2. Escalas de diagnóstico

Las escalas de diagnóstico o sistemas de puntuación clínica consisten en dar algunos valores previamente establecidos por estudios estadísticos a ciertos síntomas, signos pertinentes y relevantes [17].

En 1986, el Dr. Alvarado publicó la escala de Alvarado [1] uno de los sistemas de puntuación de apendicitis más conocidos y estudiados. La escala de Alvarado incluye tres síntomas, tres signos físicos y dos parámetros de laboratorio. Dicha escala fue nombrada MANTRELS, en representación de la primera letra delas 8 variables. A cada variable se le asigna una valoración numérica dependiendo si el paciente estudiado la presenta o no [4]. Las puntuaciones de 1-4 son negativas (No apendicitis); cuando la puntuación se encuentra entre el rango 5-6 se recomienda llevar a cabo un análisis exhaustivo del paciente mientras que las puntuaciones 7-10 se consideran positivas (Apendicitis). Ver Tabla 1.

La PAS (Pediatric Appendicitis Score, que en español se traduce como Puntuación de Apendicitis Pediátrica) es una puntuación que fue reportada por primera vez por el Dr. Samuel en la Revista de Cirugía Pediátrica en 2002. La PAS fue orientada exclusivamente a la población pediátrica, es por ello que se toma en cuenta en este estudio. Los resultados de esa investigación fueron: sensibilidad de 1, especificidad de 0.92 valor predictivo positivo de 0.96 y valor predictivo negativo de 0.99 mostrando esta herramienta como una forma sencilla pero estructurada, útil y bien fundamentada que puede mejorar la capacidad diagnostica frente a un paciente con sospecha de apendicitis [19]. Las variables y sus especificaciones se muestran en la Tabla 1.

Tabla 1. Escala de Alvarado y puntuación de apendicitis pediátrica.

Variables clínicas	Escala de Alvarado	Puntuación de apendicitis pediátrica
Migración del dolor hacia el cuadrante inferior derecho	1	1
Anorexia	1	1
Nauseas/vómitos	1	1
Hipersensibilidad en el cuadrante inferior derecho	2	2
Dolor al rebote a la palpación (Signo de Blumberg)	1	
Temperatura elevada*	1	1
Recuento de leucocitosis mayor de 10,000 por mm3	2	1
Neutrofilia mayor de 75 %	1	1
Toz/percusión/hipersensibilidad al dolor al movimiento en el cuadrante inferior derecho		2
Total	**10**	**10**
Recomendaciones		
	1-4 Apendicitis negativa 5-6 Diagnostico dudoso (Posible) 7-8 Probable apendicitis 9-10 Muy probable apendicitis	1-3 Apendicitis negativa 4-7Muy probable apendicitis 8-10 Muy probable apendicitis

* Temperatura elevada o fiebre generalmente se define como mayor o igual que 37.3° C (91. 2°F) para la escala de Alvarado Mayor o igual que 37. 3° C (99.2°F) o 38.0° C (100.4° F) para PAS.

2.3. Árboles de decisión

Los arboles de decisión es una de las técnicas de aprendizaje inductivo supervisado no paramétrico, se utiliza para la predicción y se emplea en el campo de inteligencia artificial, donde a partir de una base de datos se construyen diagramas de construcción lógica, muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que ocurren en forma repetitiva para la solución de un problema [26].

Propiedades de los Arboles de decisión.

Una de las propiedades de esta técnica es que permite una organización eficiente de un conjunto de datos, debido a que los árboles son construidos a partir de la evaluación del primer nodo (raíz) y de acuerdo a su evaluación o valor tomado se va descendiendo en las ramas hasta llegar al final del camino (hojas del árbol), donde las hojas representan clases y el nodo raíz representa todos los patrones de entrenamiento los cuales se han de dividir en clases. Los sistemas que implementan arboles de decisión tales como ID3 son muy utilizados en lo que se refiere a la extracción de reglas de dominio. Este método (ID3) se construye a partir del método de Hunt. La heurística de Hunt, consiste escoger la característica más discriminante del conjunto X, luego realizar divisiones recursivas del conjunto X, en varios subconjuntos disyuntos de acuerdo a un atributo seleccionado [26].

2.4. Algoritmo ID3

El Algoritmo ID3 fue desarrollado por J. Ross Quinlan en 1986 [22] el cual es considerado un algoritmo seminal, ya que de aquí se derivan muchos algoritmos para la construcción de árboles de decisión.

Este algoritmo se basa en la teoría de la información, desarrollada en 1948 por Claude Elwood Shannon [24]. En particular se utiliza la noción de entropía

descrita en la teoría de la información, para ver que tan aleatorio se encuentra la distribución de un conjunto de ejemplos sobre las clases a las que pertenecen [15].

El objetivo preciso del algoritmo es aprender a partir de la diferencia que existe entre los datos para analizar, esto es, un procedimiento de divide y vencerás, que maximiza la información obtenida, la cual se utiliza como una métrica para seleccionar el mejor atributo que divida los datos en clases homogéneas [3]. A continuación se muestra el pseudocódigo del algoritmo ID3. Ver Algoritmo 1.1.

Algoritmo 1.1: Pseudocódigo del algoritmo ID3 [11]

ID3 (Ejemplos[1], Atributo-objetivo[2], Atributos[3])
[1] **Ejemplos** son los datos de entrenamiento.
[2] **Atributo-objetivo** es el atributo cuyo valor va a ser predecido por el árbol y que toma valores positivos o negativos.
[3] **Atributos** es una lista con otros atributos que pueden ser ensayados o candidatos a ser elegidos para ser la raíz de este árbol.
— Inicio
— Si todos los Ejemplos son positivos, devolver un nodo etiquetado con +
— Si todos los Ejemplos son negativos, devolver un nodo etiquetado con -
— Si Atributos está vacío, devolver un nodo etiquetado con el valor más frecuente de Atributo-objetivo en Ejemplos.
— En otro caso:
– Sea A el atributo de Atributos que MEJOR clasifica Ejemplos
– Crear Árbol, con un nodo etiquetado con A.
– Para cada posible valor v de A, hacer:
- Añadir un arco a Árbol, etiquetado con v.
- Sea Ejemplos (v) el subconjunto de Ejemplos con valor del atributo A igual a v.
- Si Ejemplos (v) es vacío
* Entonces colocar debajo del arco anterior un nodo etiquetado con el valor más frecuente de Atributo-objetivo en Ejemplos.
* Si no, colocar debajo del arco anterior el subárbol **ID3** (Ejemplos (v), Atributo-objetivo, Atributos - A).
—Devolver Árbol

3. Desarrollo

3.1. Contexto de la investigación

Este trabajo fue de tipo correlacional de carácter prospectivo, longitudinal y observacional. Se llevó acabo en el Área de urgencias del Hospital General Regional "Lic. Emilio Sánchez Piedras"(Tzompantepec, Tlaxcala, México) de febrero a noviembre de 2015.

Criterios de selección Para recolectar los datos de nuestro estudio se tomaron en cuenta una serie de criterios de inclusión, exclusión y eliminación con el fin de regular nuestra muestra, Ver Tabla 2.

Tabla 2. Criterios de selección de pacientes.

Criterios de inclusión	Criterios de exclusión
- Se incluyeron todos los pacientes mayores de 4 años y menores de 15 años de edad con dolor abdominal.	- Pacientes con dolor abdominal con el antecedente de ser post operados de apendicitis.
- Pacientes con diagnóstico clínico de apendicitis efectuado por el médico en turno.	- Pacientes que tengan diagnóstico de apendicitis sin estudios de laboratorios.
- Pacientes que aceptaron participar en el estudio de forma libre y voluntaria.	- Pacientes con escala incompleta.
- Pacientes de ambos sexos.	
Criterios de eliminación	
- Pacientes que no cuenten con resultados de biopsia.	

3.2. Recopilación de datos

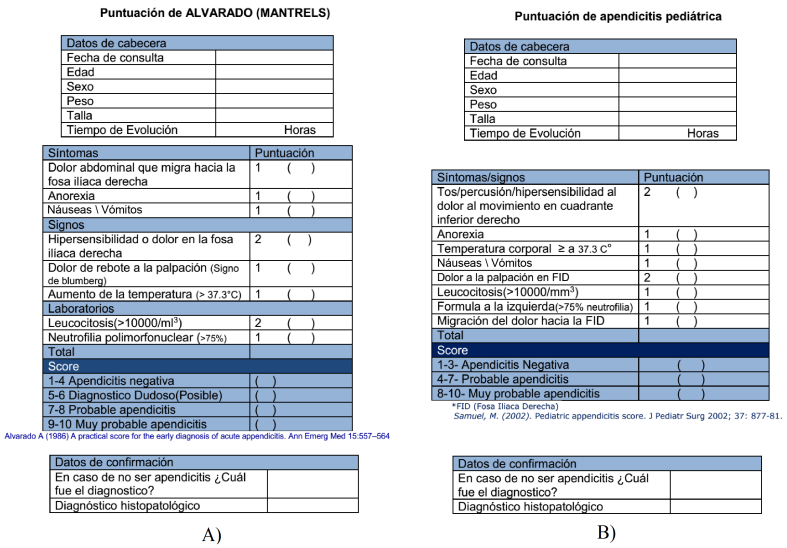
Los pacientes entre 4 y 15 años que acudieron al servicio de urgencias del Hospital General Regional *Lic. Emilio Sánchez Piedras* (Tzompantepec, Tlaxcala, México) con dolor abdominal agudo de febrero a noviembre de 2015 se incluyeron en este estudio. Pacientes con más de 15 años, con antecedentes de post operación apendicular o encuestas incompletas fueron excluidos del estudio.

La técnica de recolección de datos fue la encuesta, ya que permitió obtener los datos de modo rápido y eficaz. La Figura 1, muestra nuestras encuestas.

Estas encuestas fueron divididas en tres áreas: Datos de cabecera, datos de escala y datos de confirmación. Los datos de cabecera se utilizaron para describir a la población y el tiempo de evolución de la enfermedad. Los datos de escala se utilizaron para construir nuestros árboles de decisión ID3.

Para la selección de las variables nos basamos en la literatura de 2 escalas de puntuación diagnóstica, con el fin de determinar que puntuación tiene mejores resultados en el diagnóstico de apendicitis aguda en niños. Las escalas utilizadas fueron la escala de Alvarado [1] y Puntuación de Apendicitis pediátrica (PAS) [23].

Las variables de diagnóstico son para poder confirmar la afección y si es el caso, saber en qué fase de evolución se encontraba el miembro retirado. Además de saber el tiempo que pasan los pacientes con apendicitis aguda desde su llegada



Puntuación de ALVARADO (MANTRELS)

Datos de cabecera	
Fecha de consulta	
Edad	
Sexo	
Peso	
Talla	
Tiempo de Evolución	Horas

Síntomas	Puntuación
Dolor abdominal que migra hacia la fosa ilíaca derecha	1 ( )
Anorexia	1 ( )
Náuseas \ Vómitos	1 ( )
Signos	
Hipersensibilidad o dolor en la fosa ilíaca derecha	2 ( )
Dolor de rebote a la palpación (Signo de blumberg)	1 ( )
Aumento de la temperatura (> 37.3°C)	1 ( )
Laboratorios	
Leucocitosis(>10000/ml^3)	2 ( )
Neutrofilia polimorfonuclear (>75%)	1 ( )
Total	
Score	
1-4 Apendicitis negativa	( )
5-6 Diagnostico Dudoso(Posible)	( )
7-8 Probable apendicitis	( )
9-10 Muy probable apendicitis	( )

Alvarado A (1986) A practical score for the early diagnosis of acute appendicitis. Ann Emerg Med 15:557–564

Datos de confirmación	
En caso de no ser apendicitis ¿Cuál fue el diagnostico?	
Diagnóstico histopatológico	

A)

Puntuación de apendicitis pediátrica

Datos de cabecera	
Fecha de consulta	
Edad	
Sexo	
Peso	
Talla	
Tiempo de Evolución	Horas

Síntomas/signos	Puntuación
Tos/percusión/hipersensibilidad al dolor al movimiento en cuadrante inferior derecho	2 ( )
Anorexia	1 ( )
Temperatura corporal ≥ a 37.3 C°	1 ( )
Náuseas \ Vómitos	1 ( )
Dolor a la palpación en FID	2 ( )
Leucocitosis(>10000/mm^3)	1 ( )
Formula a la izquierda(>75% neutrofilia)	1 ( )
Migración del dolor hacia la FID	1 ( )
Total	
Score	
1-3- Apendicitis Negativa	( )
4-7- Probable apendicitis	( )
8-10- Muy probable apendicitis	( )

*FID (Fosa Iliaca Derecha)
Samuel, M. (2002). Pediatric appendicitis score. J Pediatr Surg 2002; 37: 877-81.

Datos de confirmación	
En caso de no ser apendicitis ¿Cuál fue el diagnostico?	
Diagnóstico histopatológico	

B)

Fig. 1. Herramientas para la adquisición de datos A) En base a escala de Alvarado y B) En base a Puntuación de Apendicitis Pediátrica (Encuestas).

hasta su cirugía. El diagnóstico de los pacientes fue clasificado en dos categorías: apendicitis y no apendicitis.

Estas encuestas fueron realizadas por los médicos internos de urgencias con colaboración con los médicos internos de quirófano de manera prospectiva.

Para que existiera una variabilidad en la muestra se decidió aplicar la encuesta de manera intercalada, es decir, una semana Alvarado y otra semana PAS.

Para darle veracidad al estudio se obtuvieron los resultados del análisis histopatológico de todos los miembros del estudio con cirugía realizada y se compararon con los resultados de la encuesta. El diseño del estudio fue aprobado por la junta de revisión institucional local y el consentimiento informado del paciente no era necesario.

3.3. Construcción de los árboles

Para analizar y crear el árbol de decisión ID3 utilizamos el software WEKA. Los parámetros de configuración fueron los predeterminados por el software. Para probar los resultados del clasificador se manejó el algoritmo de validación cruzada de 10 iteraciones. Weka por ser un software de análisis solo nos permite crear y evaluar el clasificador, por ello se decidió crear una interfaz gráfica con el lenguaje de programación Python con el fin de analizar de manera más exhaustiva el desarrollo de los árboles de decisión ID3.

El software generado con Python cuenta con 3 apartados principales: un módulo de importación de archivos *.csv*; una ventana de análisis de datos y

generación de reglas; y un apartado de pruebas. Como se pude ver en la Figura 2, la ventana de visualización de datos permite analizar una base de conocimientos; obtener las entropías y ganancias de manera global y de los primeros atributos del conjunto de datos. Además, generar las reglas de un árbol de decisión ID3.

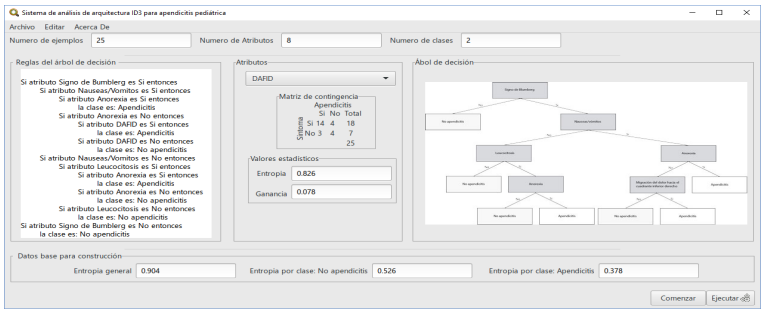


Fig. 2. Ventana de análisis de datos y generación de reglas.

En el apartado de pruebas se examina la precisión del clasificador ID3 a través de las reglas generadas. Este módulo nos permite cargar un conjunto de datos de prueba, en un archivo *.csv* y examinarlo con el clasificador generado. Este entorno nos proporciona las siguientes pruebas diagnósticas: sensibilidad, especificidad, tasas de falsos positivos, tasa de falsos negativos, el valor predictivo positivo, el valor predictivo negativo y la precisión global. Además, nos permite visualizar el resultado del análisis a través de una matriz de confusión y un gráfico de pastel que muestra el porcentaje de casos clasificados.

Para construir este software fue necesario ocupar Glade 3 [8] y las siguientes librerías externas de python: GTK+3 [9], numpy [16] y matpolib [6].

4. Resultados

Se recolectaron un total de 41 casos, 25 de ellos fueron reportados con nuestra herramienta de recolección de datos en base a la escala de Alvarado y los 16 restantes en base a la puntuación de apendicitis pediátrica. La edad promedio de la población fue de 8 años con una tendencia a variar por debajo o por encima de 4 años; nuestra población tuvo una mayor influencia en casos de varones con un 63 % contra un 37 % de mujeres; el peso promedio fue de 32.26 kilogramos con una variación de 17 kilogramos; la talla promedio fue de 124.55 centímetros con una variación de 35 centímetros; el tiempo de evolución promedio de la enfermedad hasta la atención medica fue de 35.39 horas con una variación de 67 horas; el 63 % de los paciente sufrieron automedicación antes de asistir con un médico; y el tiempo promedio de espera de un paciente desde su llegada al servicio de urgencias hasta su intervención quirúrgica de 19 casos reportados fue de 5:32 horas con una variación de 0.129 horas. En el caso de

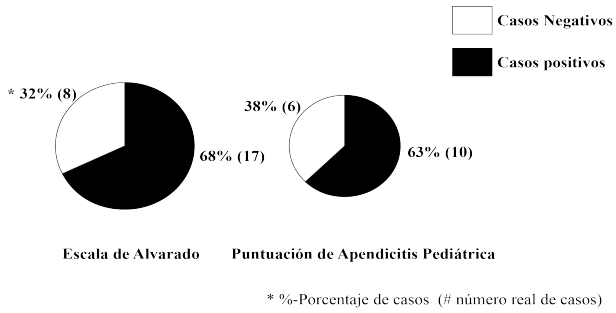


Fig. 3. Comparación de muestras recolectadas con escalas de diagnóstico médico.

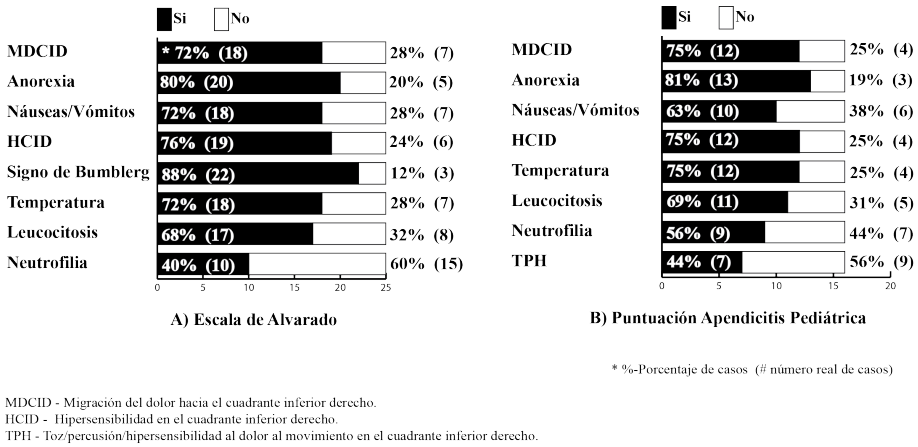


Fig. 4. Porcentaje de pacientes que presentaron signos, síntomas y laboratorios de acuerdo a cada escala de diagnóstico médico. A) Datos con respecto a la escala de Alvarado y B) Datos con respecto a la Puntuación de Apendicitis Pediátrica.

los datos recolectados con las escalas de diagnóstico médico se obtuvo un 68 % de casos positivos (Apendicitis Aguda) con la escala de Alvarado y 63 % con la Puntuación de Apendicitis Aguda, obteniendo un mayor número de casos positivos para entrenar los arboles de decisión, Ver Figura 3. La descripción porcentual de cada escala se presenta en la Figura 4. Las etiquetas (Si - No) se refieren a la presencia de dicho síntoma. En el análisis de datos se encontró que los síntomas que tenían un mayor peso en la ponderación de la escala se presentan de manera diferente en nuestra muestra. En la escala de Alvarado el síntoma signo de blumberg (88 %) e hipersensibilidad en el cuadrante inferior derecho (76 %), estas variables se encuentran en los primeros lugares de incidencia. En la puntuación de apendicitis los síntomas predominantes se presentan de manera diferente, la hipersensibilidad en el cuadrante inferior derecho tiene un valor 75 % y toz/percusión/hipersensibilidad al dolor al movimiento en el cuadrante inferior

derecho (44 %) tiene una incidencia menor dentro de los sujetos de estudio, sin embargo, el síntoma Anorexia se presenta en ambas escala con un porcentaje mayor o igual a 80 %

El resultante del árbol de decisión ID3 en base a la escala de Alvarado fue un árbol binario de 4 niveles, 13 nodos, 12 ramas, 7 hojas y profundidad 5. Los resultados para este clasificador utilizando validación cruzada de 10 iteraciones fueron: sensibilidad (0.824), especificidad (0.625), razón de probabilidad positiva (2.20), razón de probabilidad negativa (0.28), precisión global (76 %) y área bajo la curva (0.724).

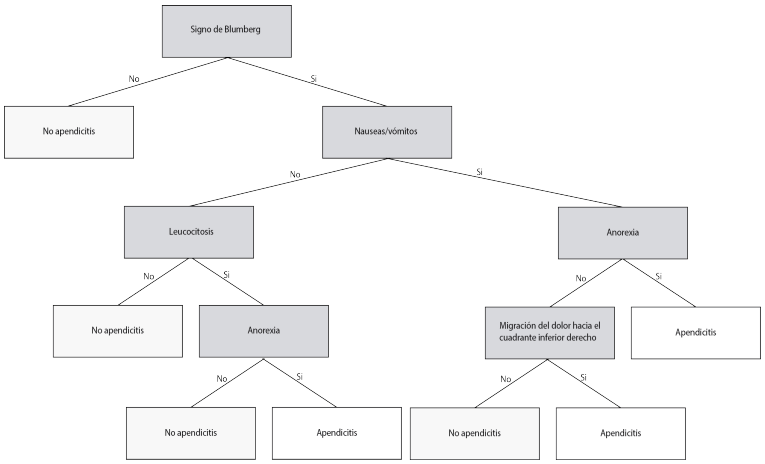


Fig. 5. Árbol de decisión ID3 en base a la escala de Alvarado.

En el caso del árbol de decisión ID3 en base a la puntuación de apendicitis pediátrica, Ver Figura 6, es un árbol binario de 2 niveles, 5 nodos, 4 ramas, 3 hojas y profundidad 3. Los resultados para este clasificador fueron los siguientes: sensibilidad (0.91), especificidad (1), razón de probabilidad positiva (0), razón de probabilidad negativa (0.09), precisión global (93.75 %) y área bajo la curva (0.917).

5. Conclusiones y trabajo futuro

La apendicitis es una enfermedad muy común en los seres humanos. A pesar de los grandes avances tecnológicos existen tazas de perforación muy altas, la capacidad de diagnóstico se complica para los especialistas en edades pediátricas y en edades avanzadas. En zonas alejadas de la ciudad se carecen de métodos de diagnóstico de muchas enfermedades entre ellas la apendicitis aguda. En este trabajo se a bordo el problema del diagnóstico de apendicitis aguda en infantes por medio de métodos computacionales inteligentes como los arboles de decisión ID3.

Tabla 3. Tabla comparativa de los trabajos relacionados

Autores	Población	Metodo	Escala de puntuación	Sensibilidad	Especificidad	País
[27]	532	Árboles de decisión C5.0	Alvarado	0.95	0.80	Taiwán
[28]	156	RNA Perceptrón multicapa/Backpropagation	Lintula	0.97	1	Turquía
[5]	100	RNA Perceptrón multicapa/Backpropagation	Alvarado	0.92	0.81	México
[20]	60	RNA Perceptrón multicapa/Backpropagation	Alvarado	1	0.97	Reino Unido
[18]	911	RNA ART1	Propia (Diferentes escalas)	P1: 0.85 P2: 0.85 P3: 0.91 P4: 0.46 P5: 0.79	P1: 0.87 P2: 0.89 P3: 0.73 P4: 0.80 P5: 0.78	Finlandia
		RNA SOM		P1: 0.83 P2: 0.88 P3: 0.62 P4: 0.55 P5: 0.55	P1: 0.90 P2: 0.77 P3: 0.82 P4: 0.87 P5: 0.83	
		RNA LVQ		P1: 0.88 P2: 0.88 P3: 0.85 P4: 0.68 P5: 0.65	P1: 0.89 P2: 0.91 P3: 0.90 P4: 0.79 P5: 0.85	
		RNA Perceptrón multicapa/Backpropagation		P1: 0.88 P2: 0.86 P3: 0.85 P4: 0.64 P5: 0.83	P1: 0.88 P2: 0.89 P3: 0.86 P4: 0.85 P5: 0.92	
[13]	180	Bosques aleatorios 200 arboles RNA Perceptrón multicapa/Backpropagation Máquinas de vectores de soporte Regresión logística	[2] con exclusión de sangre oculta en la orina y la hemoglobina.	0.94 0.94 0.91 0.91	1 0.85 1 0.62	Taiwán
[25]	2230	RNA Perceptrón multicapa/Backpropagation Clasificador bayesiano	Propia	.92 .87.		India
Sánchez et al., 2016	41	Árbol de decisión ID3	Alvarado y Puntuación de Apendicitis Pediátrica	P1=0.82 P2=0.91	P1=.63 P2= 1	México

P (Prueba) y RNA (Red Neuronal Artificial); P - Se tomó el valor menor y mayor de cada prueba.

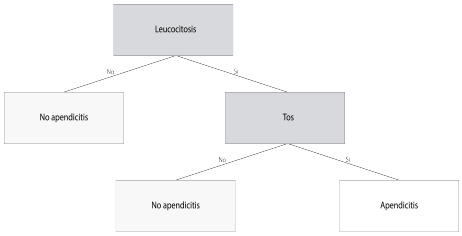


Fig. 6. Árbol de decisión ID3 en base a la puntuación de apendicitis pediátrica.

En el caso de la recolección de los datos, las escalas de diagnóstico médico fueron una elección acertada que nos permitieron reducir el número de variables clínicas y que por ende se reflejó en el número de datos a procesar. Uno de los principales problemas en este estudio fue la disposición del personal para recolectar datos, problemas como la perdida de encuestas, el no seguimiento de los casos y la atención de los pasantes hacia la investigación fueron algunas limitantes. Aunque el promedio de afecciones al año no era bastante se considera que el modelo mejoraría si se hubieran obtenido más casos.

En el desarrollo de este trabajo se consideraron diferentes plataformas para el análisis de datos, optando por Weka, una opción eficiente y eficaz para obtener modelos de clasificación, a pesar de estas características, este software carece de una herramienta para exportar las reglas del árbol de decisión ID3 y ejecutar pruebas. Con el software que se desarrolló en Python se consiguió traducir las reglas, hacer pruebas y obtener algunas métricas de validación.

En el caso de la eficiencia de los clasificadores, el mejor de los modelos fue el árbol en base a la puntuación de apendicitis pediátrica con un 91 % de sensibilidad, 100 % de especificidad, precisión global del 93.75 % y un área bajo la curva de 0.917. Es importante mencionar que aunque existe una población menor de la puntuación de apendicitis pediátrica (16 casos) se obtuvieron buenos resultados en comparación con la escala de Alvarado donde se obtuvo menor precisión. La diferencia de los clasificadores fue de 17.5 de acuerdo con su precisión global.

Como se puede ver en la tabla 3 , los trabajos que analizamos y que abordan la misma enfermedad tienen los siguientes resultados. Con la técnica de redes neuronales artificiales [28,5,20,18,13,25] tienen un desempeño que va del 87 % al 100 % de sensibilidad y en el caso de la especificidad va del 85 % al 100 %. Para el caso de los árboles de decisión y bosques aleatorios [27,13] su especificidad es de 94 % y especificidad de 80 % a 100 %. En el caso de clasificadores bayesianos [25] la especificidad es de 87.87 %. Con esto podemos determinar que nuestros clasificadores se encuentran en un rango promedio con respecto a trabajos relacionados e incluso en algunos casos superándolos en algunas métricas de eficiencia, aunque, cabe mencionar que dichas técnicas tienen otras especificaciones como el tipo de aprendizaje, es decir, RNA LVQ y SOM [18].

De acuerdo a los valores de asertividad de los árboles de decisión ID3 los resultados son prometedores y muy buenos a nivel cuantitativo. Sin embargo,

consideramos que se necesita un mayor número de ejemplos para poder certificar la calidad diagnostica de nuestro trabajo y que nos permita pasar a la siguiente fase de pruebas en humanos. Consideramos que se cumplieron los objetivos de esta investigación, es necesario seguir con este trabajo para maximizar el grado de efectividad y que pueda ser utilizada por personal médico.

Para trabajos futuros lo ideal será abarcar más hospitales del estado o cooperar con nosocomios fuera de nuestra región para obtener un número mayor de ejemplos. Además, utilizar una sola escala de diagnóstico y de esta forma no segmentar nuestra muestra. Por la aproximación que obtuvimos con nuestros datos, lo ideal sería utilizar como base la Puntuación de Apendicitis Pediátrica. Sería necesario plantear una nueva herramienta de recolección de datos pensando a futuro con el fin de probar otras técnicas como lógica difusa y de esta forma generar una herramienta multipropósito para construir con los datos recolectados diferentes modelos con diversas técnicas que nos permitan ir más allá de clasificar (Apendicitis o No apendicitis), es decir, la fase en la que se encuentra el paciente y por supuesto poder diagnosticar otras patologías con cuadro sintomatológicos parecidos a la de la apendicitis aguda, utilizando la base de conocimientos generada.

Agradecimientos. Agradecemos a la Universidad Autónoma de Tlaxcala, la Secretaria de Salud del estado de Tlaxcala, al Hospital General Regional "Lic. Emilio Sánchez Piedras"(Tzompantepec, Tlaxcala, México), al Dr. Jorge Castro Pérez (Jefe de Enseñanza del HGRESP), al Dr. Guadalupe Sánchez Pastrana (Medico Patólogo del HGT) y por supuesto a los médicos pasantes del año 2015 que nos apoyaron en la recolección de datos. Además a Fundación Telmex.

Referencias

1. Alvarado, A.: A practical score for the early diagnosis of acute appendicitis. Annals of emergency medicine 15(5), 557–564 (1986)
2. Andersson, R.: Meta-analysis of the clinical and laboratory diagnosis of appendicitis. British journal of surgery 91(1), 28–37 (2004)
3. Barrientos, E., Cruz, N., Acosta, H.: Árboles de decisión como herramienta en el diagnóstico médico. Revista Médica de la Universidad Veracruzana 9(2), 19–24 (2009)
4. Chavarría, B.C., Corral, D.M., González, S.V.: Efectividad de la tabla mantrels y sus modificaciones: detección de apendicitis en niños menores de 13 años. Avances [Pediatría] 1(1), 20–25 (2013)
5. Chavarría, B.C., Gutiérrez, S.U.: Mejoramiento del diagnóstico de apendicitis aguda en niños usando tabla de valoración mantrels a través de redes neuronales (perceptrón multi-capas). Revista digital de posgrado, investigación y extensión del Campus Monterrey (62) (Abril 2003)
6. Dale, D., Droettboom, M., Firing, E., Hunter, J.: Matplotlib. SciPy (2015), `http://matplotlib.org/contents.html`
7. Echániz, J.S., García, M.L., Ronco, M.V., Raso, S.M., Fernández, J.B., Alvarez-Buhilla, P.L.: Valor diagnóstico de la proteina c reactiva en las sospechas de apendicitis aguda en la infancia. An Esp Pediatr 48(5), 470–474 (1998)

8. GNOME-Project: Manual del diseñador de interfaces de usuario glade. GNOME-developer (2005), https://developer.gnome.org/glade/stable/
9. GNU: The python gtk+ 3 tutorial. GNU Free Documentation License 1.3 (2016), https://python-gtk-3-tutorial.readthedocs.org/en/latest/#
10. González, J., López, G., Cedillo, E., Juárez, M., González, D., López, J., González, R.: Guía de práctica clínica apendicitis aguda. Guías clínicas de la asociación mexicana de cirugía general (2014), http://amcg.org.mx/images/guiasclinicas/apendicitis.pdf
11. Gutiérrez, M., Martín, F., J, R.: Aprendizaje inductivo: árboles y reglas de decisión (2012), https://www.cs.us.es/cursos/ia2/temas/tema-02.pdf
12. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD explorations newsletter 11(1), 10–18 (2009)
13. Hsieh, C.H., Lu, R.H., Lee, N.H., andMin Huei Hsu, W.T.C., Yu-Chuan, Li, J.: Novel solutions for an old disease:diagnosis of acute appendicitis withrandom forest, support vector machines, and artificial neural networks. Surgery 149(1), 87–93 (Enero 2011)
14. Moore, K.L., Dailey, A.F., Agur, A.M.: MOORE Anatomía con orientación clinica. Edición en español, Wolters Kluwer Health (2013)
15. Moreno, B., Kinney, R.: Minería sobre grandes cantidades de datos. Master's thesis, Universidad Autónoma Metropolitana (Noviembre 2009), http://mcyti.izt.uam.mx/archivos/Tesis/Generaci%F3n2007/ICR_Benjam%EDnMoreno.pdf
16. NumPy-Community: Numpy reference. SciPy (2015), http://docs.scipy.org/doc/numpy-1.10.1/user/
17. Ohmann, C., Yang, Q., Franke, C.: Diagnostic scores for acute appendicitis. abdominal pain study group. The European journal of surgery= Acta chirurgica 161(4), 273–281 (1995)
18. Pesonen, E., Eskelinen, M., Juhola, M.: Comparison of different neural network algorithms in the diagnosis of acute appendicitis. International journal of bio-medical computing 40(3), 227–233 (Enero 1996)
19. Pinilla, A.: Aplicación de una escala diagnóstica de apendicitis en niños: estudio prospectivo. trabajo presentado para optar al título de especialista en cirugía pediátrica. universidad nacional de colombia (2013), http://www.bdigital.unal.edu.co/11423/
20. Prabhudesai, S., Gould, S., Rekhraj, S., Tekkis, P., Glazer, G., Ziprin, P.: Artificial neural networks: Useful aid in diagnosing acute appendicitis. World Journal of Surgery 32(2), 305–309 (Septiembre 2008)
21. Python: Python 2.7.11 Documentation. Python Software Foundation (2016), https://docs.python.org/2.7/
22. Quinlan, J.R.: Induction of decision trees. Machine learning 1(1), 81–106 (1986)
23. Samuel, M.: Pediatric appendicitis score. Journal of pediatric surgery 37(6), 877–881 (2002)
24. Shannon, C.E.: A mathematical theory of communication. Bell System Technical Journal 27, 379–423, 623–656 (Julio, Octubre 1984)
25. Sivasankar, E., Rajesh, R.S., Venkateswaran, S.R.: Diagnosing appendicitis using backpropagation neural network and bayesian based classifier. International Journal of Computer Theory and Engineering 1(4), 358–363 (Octubre 2009)
26. Solarte, G., Soto, J.: Arboles de decisiones en el diagnóstico de enfermedades cardiovasculares. Scientia et Technica 3(49), 104–109 (2011)

27. Ting, H.W., Wua, J.T., Chan, C.L., Lin, S.L., Chen, M.H.: Decision model for acute appendicitis treatment with decision tree technology. a modification of the alvaro scoring system. Journal of the chinese medical association 73(8), 401–406 (Agosto 2010)
28. Yoldas, O., Mesut, T., Karaca, T.: Artificial neural networks in the diagnosis of acute appendicitis. The american journal of emergency medicine 30(7), 1245–1247 (Septiembre 2012)

Técnicas de aprendizaje automático aplicadas a electroencefalogramas

Omar Santa-Cruz, Lorena del Mar Ramírez, Felipe Trujillo-Romero

Universidad Tecnológica de la Mixteca,
Huajuapan de León, Oaxaca,
Mexico

iccsantacruz@gmail.com, lorenadelmar.rmz@gmail.com,
ftrujillo@mixteco.utm.mx

Resumen. Los electroencefalogramas(EEG) almacenan la actividad cerebral en un instante de tiempo determinado. Actualmente los EEG son utilizados en disciplinas médicas, científicas y tecnológicas. Debido a esto, lograr interpretar y clasificar de forma precisa estas señales es un problema abierto en el área de machine learning. En esta investigación se trabaja con un conjunto de datos que representan señales de EEG, en las cuales se encuentran codificados los estados de los ojos (abiertos o cerrados). Este conjunto ha sido analizado por diversos autores, lo que se propone en esta investigación es tratar de mejorar el desempeño de los métodos ya explorados. Por lo anterior, se explora mediante una búsqueda greedy los valores de los parámetros que ofrezcan un mayor rendimiento al utilizar SVM y RNA. También se aplican técnicas de reducción mediante selección de prototipos y extracción de características al conjunto de señales. El objetivo es tratar de mejorar la estructura de los datos mediante una representación más compacta, para posteriormente comprobar si se logra mejorar el proceso de discriminación.

Palabras clave: EEG, SVM, redes neuronales artificiales, extracción de características, selección de prototipos, optimización, predición del los estados de los ojos.

1. Introducción

Las nuevas tecnologías en el ámbito médico y su fusión con áreas de las ciencias de la computación han dado origen a la llamada informática médica. La implementación de software que colabore en el diagnóstico médico es un problema de investigación con gran demanda en esta área[1]. Bioinformática y química computacional son dos claros ejemplos de la relación de la computación con áreas en donde la cantidad de datos que se procesa durante una investigación rebasa la capacidad de proceso humano. Un ejemplo de diagnóstico médico asistido por computadora son los sistemas CADx (Computer-Aided Diagnosis) y CAD (Computer Aided Detection)[2,3], los cuales se utilizan para ayudar a los radiólogos en diagnóstico y detección de anormalidades en imágenes de

mamografías. Existen otros tipos de sistemas automáticos de reconocimiento de patrones que auxilian en el diagnóstico diferencial de casos médicos, los cuales involucran electrocardiogramas, electroencefalogramas, radiografías, tomografías, entre otros [1,4,6]. El diagnóstico médico objetivo y más acertado depende de la experiencia que el especialista ha adquirido a través de los años al tratar cientos de casos. Sin embargo, debido al número de variables que se deben manejar (edad, sexo, el tiempo de evolución, síntomas específicos) en ciertos casos clínicos, la probabilidad de error del diagnóstico diferencial aumenta. La hipótesis médica puede verse afectada por el sesgo de confirmación debido a muchos factores como son: estado de ánimo, falta de experiencia, influencia médica externa, etc. La incursión del aprendizaje automático en el área médica puede ayudar a la consolidación de un sistema de diagnóstico más eficiente [1,5]. El uso de un software que apoye al especialista médico a tomar una decisión debe empezar a considerarse como una necesidad básica debido al manejo de datos en grandes cantidades y con relaciones por lo general muy complejas entre ellos. Un programa de diagnóstico médico puede igualar la capacidad de un experto humano, además, no sólo es capaz de reconocer un patrón clínico y dar un diferencial, sino también puede facilitar el trabajo del médico, además de reforzar sus conocimientos, y establecer un pronóstico de riesgo [1,6]. En este documento se explora un problema de predicción de los estados de los ojos (abiertos o cerrados) mediante información extraída por medio de electroencefalogramas (EEG). Los EEG son ampliamente utilizados en muchos problemas de medicina para diagnosticar diferentes enfermedades.

2. Electroencefalogramas

Un EEG es un conjunto de señales que proviene de la captura de actividad cerebral por medio de sensores específicamente diseñados para esto. Un electroencefalograma registra las diferencias de potencial eléctrico producidas en el cerebro cuando se realiza cierta actividad[8]. Este potencia eléctrico en forma de impulsos contiene información sobre la comunicación neuronal en el cerebro. En la forma de onda de la señal se encuentra almacenada la información sobre las intenciones del usuario en un momento determinado. Por medio del análisis y búsqueda de patrones con técnicas de machine learning es posible obtener un modelo que logre predecir ciertos estados de la mente[9]. Algunos ejemplos de aplicaciones en las que es útil la información de un electroencefalograma son: reconocimiento de emociones[9,33], realidad virtual[14,15,16], videojuegos[17], interfaces cerebro-computadora[10,11], en el ámbito militar[13,32] y en software de control de objetos para personas con algún tipo de discapacidad [11,12]. También es posible predecir enfermedades que causan degeneración de células cerebrales como lo hace el Alzheime's [18].

En este contexto, reconocer estímulos con suficiente precisión es de vital importancia en los problemas antes mencionados. Detectar el estado de los ojos (abiertos o cerrados) puede ser de utilidad en sistemas de prevención de accidentes [19], identificación de trastornos de sueño infantil[20], detección de convul-

siones epilépticas[21], clasificación de desorden bipolar y detección de pacientes con desorden de déficit de atención [22], identificación de las características de estrés [23] y detección de parpadeo en el ojo humano[24]. El problema de clasificación de los estados de los ojos se ha atacado con diferentes métodos de aprendizaje automático. Por ejemplo, Fukuda et al.[25], utilizan una red neuronal artificial para la clasificación del estado del los ojos. Yeo et al.[19] se enfocan en la detección de somnolencia durante la conducción de vehículos por medio del estado de los ojos, para ello, utiliza una máquina de soporte vectorial. Polat y Gne [21] utilizan un árbol de decisión en la detección de crisis epiléptica . Sulaiman et al. [23] utilizan k-NN para la identificación de características de estrés . Además, Rsler y Suendermann [26] generan un corpus con cerca de $15,000$ muestras y someten los datos a 42 clasificadores auxiliándose de la plataforma WEKA[29]. [27,28,30,31] utilizan el mismo corpus([26]) para aplicar otros enfoques de clasificación a los ya reportados.

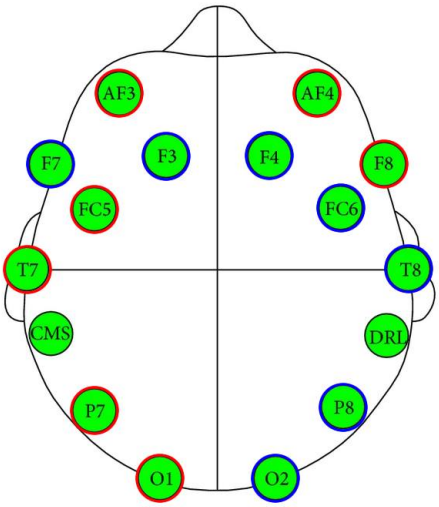


Fig. 1. Esquema general de la posición de los sensores. Los sensores marcados con color azul toman valores máximos, mientras que los sensores marcados en color rojo toman los valores mínimos cuando los ojos estan abiertos [29].

Es este trabajo se analiza el mismo corpus, se utiliza SVM y una RNA con una búsqueda greedy que logre la obtención de los mejores parámetros del clasificador. Además se utiliza un algoritmo de selección de instancias y de extracción de características para reducir el número de muestras y el número de variables en el conjunto de datos respectivamente. Como prueba final, los conjuntos reducidos se someten a los clasificadores antes mencionados.

3. Algoritmos de aprendizaje automático

3.1. Prototype selection by clustering (PSC)

En un conjunto de entrenamiento T, los prototipos de borde, son relevantes para encontrar la frontera de las distintas clases, estos prototipos que se encuentran en la frontera de las clases son siempre críticos para el proceso de clasificación. Las instancias localizadas en el interior de estas regiones puede ser considerados como superfluas ya que su eliminación no conduce a ninguna perdida en la capacidad de encontrar el discriminante de clase [34].

PSC utiliza C-means para dividir el conjunto de datos en C-clusters. Después de realizar el agrupamiento se analiza y se identifica cual de ellos es un cluster homogéneo y cual no lo es. Para los clusters no homogéneos se detectan los prototipos de borde de las diferentes clases y se elimina el resto. Para las regiones homogéneas PSC conservar sólo al prototipo que mejor represente a la región [36], esto es, la instancia más cercana a la media del grupo.

La forma en que PSC decide que prototipos son de borde, es subdividiendo los cluster no homogéneos de acuerdo a su clase de pertenencia. El subgrupo con mayor número de elementos es marcado como el subgrupo de clase mayoritaria [37]. Una vez que el subconjunto de clase mayoritaria ha sido encontrado, los prototipos de borde de los diferentes subgrupos de cada clusters serán los vecinos más cercanos(1-NN) a algún prototipo que pertenezca a la clase mayoritaria. De forma análoga, PSC encuentra los prototipos de borde del subgrupo de clase mayoritaria buscando a los prototipos que sean los vecinos más cercanos del algún prototipo de clase mayoritaria, pero que no pertenezca a dicha clase[38]. De forma análoga se encuentran los prototipos de borde de los subgrupos restantes.

3.2. Máquinas de soporte vectorial(SVM)

Si T representa un conjuntos de m datos de dimensión n en el que cada ejemplo x_i esta asociado a una etiqueta $y_i \in [-1, 1]$. El hiperplano que separa a ambas clases es un vector que satisface la siguiente ecuación :

$$w \cdot x - b = 0 \tag{1}$$

donde w representa la norma del vector al hiperplano y $\frac{b}{\|w\|}$ la distancia perpendicular al origen. SVM puede encontrar los valores óptimos que corresponda al hiperplano de margen máximo minimizando $\|w\|^2$ y garantizando que se cumpla la siguiente expresión: $y_i(x_i * w + b) - 1 \geq 0$, $\forall i$, donde y_i es la etiqueta asociada a la instancias x_i[42]. Para resolver este problema de optimización con restricción se puede utilizar los multiplicadores de Lagrange en donde la función objetivo L_p (lagraniano primal)puede ser minimizada con respecto a w y b:

$$L_p(w, b, \alpha) = \frac{1}{1} \|w\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(x_i * w + b) - 1) \tag{2}$$

Esto quiere decir que se puede resolver el problema dual asociado al problema primal, que equivale a replantear la solución de la siguiente forma:

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i * x_j \tag{3}$$

donde w, b desaparecen y los vectores del conjunto de entrenamiento que proporcionan un multiplicador $\alpha_i > 0$ son denominados vectores de soporte. El algoritmo SVM fue modificado en [46], introduciendo el concepto de margen suave, el cual permite que el margen no sea tan estrícto, agregando una constante C al error de clasificación ϵ. SVM puede ser extendido a problema de separación no lineal aplicando el llamado truco kernel. Una particularidad de las funciones kernel es que el producto punto de las imágenes en $\Phi(X)$, el cual representa un mapeo no lineal, es equivalente a evaluar los datos del conjunto original en la función kernel, esto es:

$$\langle \Phi(x_i), \Phi(x_j) \rangle = k(x_i, x_j) \tag{4}$$

donde $\langle , \rangle$ representa el producto punto.

3.3. Redes neuronales artificiales (RNA)

Una red neuronal recibe como entrada un conjuntos de $i = 1, ...n$ patrones en forma de vectores de dimensión p, cada vector de entrada es procesado a través de las neuronas de las I capas ocultas de acuerdo a las conexiones entre los nodos de estas. Cada nodo contiene un función de activación f, la cual mediante una suma ponderada de las entradas de los nodos y un valor de sesgo adicional obtiene el valor de salida del nodo[40].

Existen diferentes tipos de arquitectura, interconexiones y algoritmos de aprendizaje para la creación de una red neuronal. La estructura de una red neuronal de tipo feedforward recibe ese nombre debido a que las conexiones de sus neuronas fluyen unidireccionalmente desde la capa de entrada hasta la capa de salida sin que en el proceso haya ciclos de retroalimentación entre las neuronas. La regla más utilizada para que la red aprenda es el algoritmo backpropagation. El algoritmo de backpropagation utiliza la técnica del gradiente descendente para buscar el mínimo de la función de error en el espacio de pesos de los nodos. Así, mediante la obtención del gradiente con respecto a los pesos de la red se calcula la combinación de pesos que minimice dicha función[42].

Suponiendo que la integral de la función de activación es sólo la suma de las entradas del nodo, y debido a que la salida de los nodos en cada capa se pueden ver como una aplicación sucesiva (función composición) de la función de salida de los nodos de la capa anterior; es posible la obtención del gradiente de la función mediante cualquier función de transferencia diferenciable f.

La ecuación del error general de la red es:

$$E = \frac{1}{2} \sum_{i=1}^{m} \|o_i - t_i\|^2 \tag{5}$$

Conociendo el error E se puede calcular el gradiente con respecto de los pesos asociados a cada nodo y posteriormente reajustar los pesos y los bias de la red. El proceso de optimización termina cuando el error general de la red converge o bien cuando el error alcanza un valor mínimo deseado.

Gradiente descendente con momentum Cuando el mínimo de la función de error en el espacio de pesos se encuentra en una 'superficie' r-dimensional estrecha y la tasa γ no es la adecuada, esto provoca que la convergencia del error oscile, Lo cual genera un retardo significativo de la convergencia y en el proceso de aprendizaje. Si utilizamos una constante de momentum β, que funcione como un filtro pasa bajas podemos evitar oscilaciones excesivas llegando a la convergencia de la red en un menor número de iteraciones. No existe una regla general para el ajuste de los dos parámetros de aprendizaje que acelere al algoritmo en la convergencia, normalmente este ajuste se realiza mediante ensayo y error o mediante una búsqueda greedy.

Gradiente descendente con tasa de aprendizaje variable Este algoritmo de aprendizaje trata de mantener la constante de aprendizaje γ lo más alta posible, haciendo que γ varíe de forma proporcional a la complejidad de la superficie de error. Al acoplarse a la forma de la superfície estabiliza el aprendizaje durante todo el proceso de entrenamiento. Existen diferentes técnicas para lograr que la tasa de aprendizaje varíe de acuerdo a rendimiento de la red para cada iteración. De forma general la lógica a seguir es la siguiente: Si el error disminuye, la dirección del gradiente es la correcta y se puede acelerar el proceso de convergencia incrementando el tamaño de γ. Por el contrario, si el error se incrementa el algoritmo actúa decrementando la tasa de aprendizaje[41,42]. El algoritmo Backpropagation con momentum y aprendizaje variable garantizan rápida convergencia para algunos problemas, sin embargo el valor adecuando de todos los parámetros deber ser asignado mediante ensayo y error, no garantizando la convergencia para problemas muy complejos.

3.4. Análisis de componentes principales (ACP)

Dado un conjunto de datos con p variable, ACP transforma dicho conjunto en uno nuevo con r variables, siendo $r << p$. Este nuevo conjunto debe cumplir ciertas restricciones: las nuevas variables, las cuales reciben el nombre de componentes principales, deben ser una combinación lineal de las variables de partida, además, cada nueva variable debe representar una parte de la variabilidad de los datos originales. Cuanto mayor sea la varianza de las componentes, mayor es la información que almacenan las nuevas variables. Siguiendo esta lógica, la varianza debe ir decreciendo según se vaya obteniendo cada nueva variable. Cabe mencionar que la suma total de estas varianzas es igual a la suma de las varianzas del conjunto original [44]. Otra condición que debe cumplir el nuevo conjunto es que las nuevas variables no estén correladas, ya que esta correlación impide apreciar de forma precisa el rol que juega cada variable en el problema bajo investigación [43].

4. El conjunto de datos

El conjunto de datos que se emplea en los experimentos es obtenido del repositorio de la Universidad de California (UCI) [39]. El conjunto de datos consta de 14980 ejemplos, que contiene 14 características descriptivas, donde cada característica representa la informació que recolecta cada uno de los 14 sensores como muestra la figura 2. Cada medición dura 117 segundos, las señales que emiten las ondas cerebrales son retenidas por los sensores y mediante un proceso de transducción son mapeados a valores reales. El estado de los ojos asociado a cada secuencia se añadió de forma manual, después de analizar los fotogramas de vidéo que captó una cámara durante todo el proceso de obtención de la información. Cada ejemplo del conjunto de muestras tiene asociada las clase 1 cuando el ojo se encuentra abierto y la clase 0 de forma contraria. En [26] se analizan los rangos máximos y mínimos de las variables para los dos estados del ojos. En este análisis se concluye que existe una diferencia significativa entre los dos estados (abiertos y cerrados) que permite realizar una discriminación mediante algoritmos de aprendizaje automático.

5. Método propuesto

Lo que se propone en esta investigación es tratar de mejorar la precisión de clasificación de algunas técnicas reportadas en el estado del arte, específicamente se trata de mejorar la precisión de SVM en [27] y del perceptrón multi-capa utilizado en [26,31]. Además se pretende verificar la precisión de k-NN debido a que en [26,27,28,31] se reporta una precisión de clasificación variable para 1-NN al utilizar WEKA. Por otra parte se someterá al conjunto de datos a un ACP, la cual es una técnica de extracción de características basada en la varianza, después de aplicar ACP se pretende verificar si k-NN, SVM y RNA logran mejorar la precisión reportada en la literatura utilizando el conjunto de datos original. De forma análoga, se aplica PSC, la cual es una técnica de selección de instancias basada en k-NN y C-means. Se utiliza ACP debido a que en las gráficas de dispersión por pares se observa correlación entre algunas variables. Desde otra perspectiva, PSC fue seleccionado debido a que k-NN es uno de los algoritmos que genera mayor rendimiento en el conjunto de datos original. La figura 2 muestra el diagrama de flujo del proceso planteado.

6. Preliminares

Para la etapa de experimentos se utilizó MATLAB R2013a, la plataforma WEKA 3,6 y S.O. Ubuntu 14,04 a 64 bits. El hardware utilizado fue una máquina de escritorio con procesador Phenom $X4$ a 2,5 GHz, 8 Gb de RAM. Para las pruebas de precisión se utilizó validación cruzada estratificada con 10 particiones. Para la búsqueda greedy de los diversos parámetros de SVM y RNA se seleccinó de forma aleatoria el 50 % del conjunto de datos. Para obtener el mejor modelo de PSC se utilizó validación cruzada estratificada. Para cada iteración se aplicó PSC

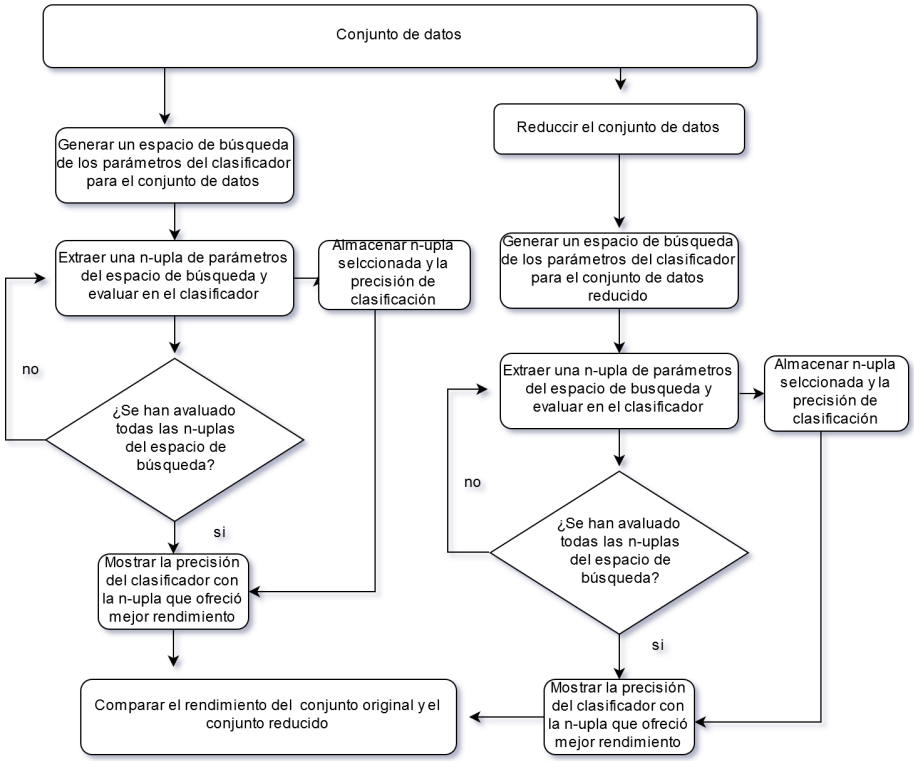


Fig. 2. Diagrama de flujo del método propuesto.

al conjunto de entrenamiento y se validó con el conjunto de prueba con 1-NN. El subconjunto con el que se obtuvo el mejor rendimiento para las 10 iteraciones se seleccionó y se descartaron las instancias restantes, tanto del conjunto de prueba como del conjunto de entrenamiento. Para las pruebas de clasificación sólo se ocupo el subconjunto seleccionado.

La primera interacción con los datos fue durante el análisis exploratorio, donde se observó su distribución, la posible correlación entre variables y se eliminaron 4 datos atípicos. El conjunto de datos final cuenta con $6,722$ casos de los ojos cerrados y $8,254$ casos de ojos abiertos. En [26,27,28,31] se reporta que la mejor precisión de clasificación es obtenida con k-NN y KStar, ambos clasificadores basados en instancias. [27] aborda el problema mediante máquinas de soporte vectorial, no obteniendo los resultados deseados, en [31] utilizan k-NN y un perceptrón multicapa para clasificar el conjunto, k-NN obtiene los mejores resultados. Las primeras pruebas se realizan con k-NN y KStar , obteniendo resultados similares, 1-NN obtuvo una precisión de clasificación de 97,42, mientras que para KStar su precisión es de 96,66.

7. Experimentos y resultados

El siguiente clasificador elegido para las pruebas es SVM, mediante una búsqueda greedy, se optimizaron los parámetros $C = 2^{-5}, ..., 2^{10}$ y $gamma = 2^5, ..., 2^{-10}$ para los 4 kernels que incluye la librería LIBSVM [45] de MATLAB. Los mejores resultados fueron para el kernel RBF y kernel polinomial con una precisión de clasificación de 98,865 y 93,095 respectivamente. Los parámetros utilizados son: $C = 13{,}4543$ y $gamma = 0{,}0017074$ para el kernel RBF y para el kernel polinomial $C = 0{,}42045$ y $gamma = 0{,}005309$. Las pruebas de clasificación utilizando una red neuronal feedforward y el algoritmo de backpropagation con momentum y tasa de aprendizaje variable se realizaron en MATLAB. Primero se realizó una búsqueda greedy para obtener los parámetros iniciales más adecuados para la constante de momemtum, el número de capas ocultas, el número de neuronas por capa, la tasa de aprendizaje, el incremento de la tasa de aprendizaje y el decremento de la misma. La tabla 1 muestra el mejor resultado de la búsqueda.

Tabla 1. Valores de los parámetros de la red neuronal.

Parámetro	Mejor valor
Tasa de aprendizaje	0,005
Incremento del aprendizaje	0,0675
Decremento del aprendizaje	0,255
Momentum	0,63
Núm. épocas	50000
Configuración de la red	$[14, 7, 4, 2, 1]$
Precisión	88,67

Debido a que el número de muestras del conjunto de datos se acerca a $15,000$ intancias con 14 características por muestras, los experimentos siguientes se enfocan a reducir el número de instancias y en número de características del conjunto de datos para observar el comportamiento de la precisión de clasificación con los algoritmos utilizados anteriormente.

La mejor precisión de clasificación reportada en el estado del arte se obtiene con $k-$NN y KStar, por este motivo se utiliza el algoritmo PSC. Este algoritmo reduce el conjunto de entrenamiento mediante clusters y k-NN. El parámetro c que recibe PSC es el número de clusters a crear. En [38] se menciona que el número de clusters recomendado es $c = f * l$, con $f = 10$, donde l representa en número de clases y f una constante de intensidad. Para las pruebas de reducción con PSC se hizo variar el parámetro $f \in [1, 8000]$. Para la obtención del modelo que mejor generalize el proceso de discriminación de nuevas muestras se utilizó k-NN . El mejor modelo creado por PSC obtuvo un rendimiento de 96,79 para 1-NN. Los detalles de este modelo son los siguientes:

Tabla 2. Valores de los parámetros para el mejor modelo obtenido por PSC.

Parámetro	valor
Núm. de clusters	6000
Umbral de k-meas	1
Núm. de instancias finales	$6,275$
% de retención	41,90
Rendimiento en el entrenamiento	97,80
Parámetro k	1

Para KStar el rendimiento con este subconjunto es de 94,659, para el mejor modelo de la red neuronal el rendimiento es 88,74 y para SVM con RBF se obtiene una precisió de 98,67.

Por último, se sometió al conjunto de datos original a un análisis de componentes principales, del cual se tomaron los primeros 7 componentes. Para los 7 primeros componentes principales, la retención de la varianza es de 91,03 %. Al someter la nueva representación de los datos a los clasificadores el rendimiento se decremento de forma considerable comparado con los resultados anteriores. Estos resultados no son los esperados debido a que en las gráficas de dispersión por pares se observaba una aparente correlación entre algunos pares de variables. La siguiente tabla(ver tabla 3) resume el rendimiento obtenido para los 4 algoritmos de clasificación aplicado a este subconjunto.

Tabla 3. Tabla de precisión de clasificación para el subconjunto de datos obtenido al aplicar ACP.

Clasificador	Precisión
1-NN	86,879
KStar	86,99
RNA ($[7, 4, 2, 1]$)	72,96
SVM (RBF, C = 1, gamma = 1,992)	89,95

8. Conclusiones y trabajo a futuro

En este investigación se logra comprobar que las SVM ofrece resultados superiores a los reportados en el estado del arte cuando los parámetros del algoritmo son los adecuados. También podemos constatar que k-NN y KStar son una muy buena opción para las discriminación de los estados de los ojos. Otro punto a favor es que ofrecen resultados de rendimiento similares cuando el conjunto es reducido por PSC. Es válido mencionar que PSC es una buena elección para reducir un conjunto de datos en el que los mejores resultados de precisión los obtienen clasificadores basados en distancias. Las redes neuronales no ofrecieron un rendimiento similar al de los demás clasificadores utilizados,

pero si son superiores a los reportados en [31] con el perceptrón multicapa. Para la RNA es posible mejorar su precisión de clasificación si los parámetros de la red logran ser optimizados mediante otro enfoque, ya que una búsqueda greedy no es la mejor opción. Por otro lado, ACP no ofreció los resultados esperados, pero se puede notar que el rendimiento de SVM sigue siendo superior, seguido por k-NN y KStar.

Como trabajo a futuro se plantea mejorar el desempeño de la red optimizando sus parámetros y topología mediante un algoritmo genético. Los algoritmos genéticos pueden ser capaces de encontrar una mejor configuración para el número de capas, número de nodos por capa, conexiones entre los nodos, conexiones entre las capas, pesos y funciones de activación de los nodos. También se plantea experimentar con otros algoritmos de selección de instancias y de extracción de características esperando obtener una mejor estructura de los datos en una forma más compacta.

Referencias

1. Lugo-Reyes, S.O., Maldonado-Colín, G., Murata, C.: Inteligencia artificial para asistir el diagnóstico clínico en medicina. Revista Alergia, 61:110–120 (2014)
2. Bozek, J., Mustra, M., Delac, K., Grgic, M.: A survey of image processing algorithms in digital mammography. Recent Advances in Multimedia Signal Processing and Communications, 231, pp. 631–657 (2009)
3. Cheng, H. D., Shi, X., Min, R., Hu, L., Cai, X., Du, H. N.: Approaches for automated detection and classification of masses in mammograms. Pattern Recognition , Vol. 39, No. 4, pp. 646–668 (2006)
4. Murata, C., Ramírez, A.B., Ramírez, G., Cruz, A. et al.: Análisis discriminante para predecir el diagnóstico clínico de inmunodeficiencias primarias: reporte preliminar. Revista Alergia, 62:125–133 (2015)
5. Khattree, R.: Computational methods in biomedical research. Baton Rouge, LA: Chapman & Hall (2007)
6. Cleophas, T.J., Zwinderman, A.H.: Machine Learning in Medicine. Springer (2013)
7. Wang, T., Guan, S.-U., Man, K.L., Ting, T.O.: EEG Eye State Identification Using Incremental Attribute Learning with Time-Series Classification. Mathematical Problems in Engineering, Article ID 365101, 9 pages (2014)
8. Pour, P., Gulrez, T., Al Zoubi, O., Gargiulo, G., Calvo, R.: Brain Computer Interface: Next Generation Thought Controlled Distributed Video Game Development Platform. Proc of the CIG, Perth, Australia (2008)
9. Zheng, W.L., Dong, B.N., Lu, B.L.: Multimodal emotion recognition using EEG and eye tracking data. Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE, pp. 5040–5043 (2015)
10. Taylor, D.M., Tillery, S.I.H., Schwartz, A.B.: Direct Cortical Control of 3D Neuroprosthetic Devices. Science 296, pp. 1829–1832 (2002)
11. Lebedev, M.A., Nicolelis, M.A.L.: Brain-Machine Interfaces: Past, Present and Future. Trends in Neurosciences, Vol. 29, No. 9, pp. 536–546 (2006)
12. Wang, Y. & Makeig, S.: Predicting Intended Movement Direction Using EEG from Human Posterior Parietal Cortex. Dylan Schmorrow; Ivy V. Estabrooke & Marc Grootjen, eds., 'HCI (16)' , Springer, pp. 437–446 (2009)

13. Muzik, J., Hana, K.: Real-time BSPM processing system, 4th European Conference of the International Federation for Medical and Biological Engineering, Vol. 22, pp. 377–380 (2009)
14. Doud, A.J., Lucas, J.P., Pisansky, M.T., He, B.: Continuous three-dimensional control of a virtual helicopter using a motor imagery based brain-computer interface. PLoS One 6(10) (2011)
15. McFarland, D.J., Sarnacki, W.A., Wolpaw, J.R.: Electroencephalographic (EEG) control of three-dimensional movement. J Neural Eng 7:036007 (2010)
16. Royer, A.S., Doud, A.J., Rose, M.L., He, B.: EEG control of a virtual helicopter in 3-dimensional space using intelligent control strategies. IEEE Trans Neural Syst Rehabil Eng, Vol. 18, No.6, pp. 581589 (2010)
17. Navalyal, G.U., Gavas, R.D.: A dynamic attention assessment and enhancement tool using computer graphics. Navalyal and GavasHuman-centric Computing and Information Sciences, Vol. 4, No. 11, Springer (2014)
18. Dauwels, J., Vialatte, F., Cichocki, A.: Diagnosis of Alzheimer's disease from EEG signals: where are we standing? Vol. 7, No. 6, pp. 487–505 (2010)
19. Yeo, M.V.M., Li, X., Shen, K., Wilder-Smith, E.P.V.: Can SVM be used for automatic EEG detection of drowsiness during car driving? Safety Science, Vol. 47, No. 1, pp. 115–124 (2009)
20. Estévez, P.A., Held, C.M., Holzmann, C.A., Perez, C.A., Pérez, C.A., Heiss, J., Garrido, M., Peirano, P.: Polysomnographic pattern recognition for automated classification of sleep-waking states in infants. Medical and Biological Engineering and Computing, Vol. 40, No. 1, pp. 105–113 (2001)
21. Polat, K., Gne, S.: Classification of epileptic form EEG using a hybrid system based on decision tree classifier and fast Fourier transform. Applied Mathematics and Computation, Vol. 187, No. 2, pp. 1017–1026 (2007)
22. Sadatnezhad, K., Boostani, R., Ghanizadeh, A.: Classification of BMD and ADHD patients using their EEG signals. Expert Systems with Applications, Vol. 38, No. 3, pp. 1956–1963 (2011)
23. Sulaiman, N., Taib, M.N., Lias, S., Murat, Z.H., Aris, S.A.M., Hamid, N.H.A.: Novel methods for stress features identification using EEG signals. International Journal of Simulation: Systems, Science and Technology, Vol. 12, No. 1, pp. 27–33 (2011)
24. Nguyen, T., Nguyen, T.H., Truong, K.Q.D., van Vo, T.: A mean threshold algorithm for human eye blinking detection using EEG. Proceedings of the 4th International Conference on the Development of Biomedical Engineering in Vietnam, pp. 275–279, Ho Chi Minh City, Vietnam (2013)
25. Fukuda, O., Tsuji, T., Kaneko, M.: Pattern classification of EEG signals using a log-linearized Gaussian mixture neural network. Proceedings of the IEEE International Conference on Neural Networks. Part 1 (of 6), pp. 2479–2484, Perth, Australia (1995)
26. Rösler, O., Suendermann, D.: First step towards eye state prediction using EEG. Proceedings of the International Conference on Applied Informatics for Health and Life Sciences (AIHLS '13), Istanbul, Turkey (2013)
27. Jain, N., Bhargava, S., Shivani, S., Goyal, D.: Eye state prediction using EEG by supervised learning. International Journal of Science, Engineering and Technology (2015)
28. Sahu, M., Nagwani, N.K., Verma, S., Shirke, S.: Performance Evaluation of Different Classifier for Eye State Prediction Using EEG Signal. International Journal of Knowledge Engineering, Vol. 1, No. 2 (2015)

29. Hall, M. Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter, Vol. 11, pp. 10–18 (2009)
30. Wang, T., Guan, S.-U., Man, K.L., Ting, T.O.: EEG Eye State Identification Using Incremental Attribute Learning with Time-Series Classification. Mathematical Problems in Engineering.
31. Sabanc, K., Koklu, M.: The Classification of Eye State by Using k-NN and MLP Classification Models According to the EEG Signals. International Journal of Intelligent Systems and Applications in Engineering, pp.127–130 (2015)
32. van Erp, J., Reschke, S., Grootjen, M., Brouwer, A.-M.: Brain Performance Enhancement for Military Operators. HFM Symposium on Human Performance Enhancement for NATO Military Operations (Science and Technology and and Ethics) (2009)
33. Pham, T., Tran, D.: Emotion recognition using the emotiv epoc device. Lecture Notes in Computer Science, Vol. 7667 (2012)
34. Kuncheva, L.I., Bezdek, J.C.: Nearest prototype classification, clustering, genetic algorithms, or random search? IEEE Trans Syst Man Cybern C28(1), pp. 160–164 (1998)
35. Bezdek, J.C., Kuncheva, L.I.: Nearest prototype classifier designs: an experimental study. Int J Intell Syst, Vol. 16, No. 12, pp. 1445–1473 (2001)
36. Wilson, D.R., Martínez, T.R.: Reduction techniques for instance-based learning algorithms. Mach Learn, Vol. 38, pp. 257–286 (2000)
37. Brighton, H., Mellish, C.: Advances in instance selection for instance-based learning algorithms. Data Min Knowl Disc, Vol. 6, No, 2, pp. 153–172 (2002)
38. Olvera-López, J.A., Carrasco-Ochoa, J.A., Martínez-Trinidad: A new Fast Prototype Selection Method based on Clustering. Pattern Analysis and Applications. Pattern Analysis and Applications, Vol. 13, No. 2, pp. 131–141 (2010)
39. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
40. Rojas, R.: Neural Networks: A Systematic Introduction. Springer-Verlag New York, Inc., New York, NY, USA. (1996)
41. Moreira, M., Fiesler, E.: Neural Networks with Adaptive Learning Rate and Momentum Terms. Technical report, Institut Dalle Molle d'intelligence artificielle perceptive (1995)
42. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall (1998)
43. Lee, J.A., Verleysen, M.: Nonlinear Dimensionality Reduction. (1st ed.) Springer Publishing Company, Incorporated (2007)
44. Venna, J., Peltonen, J., Nybo, K., Aidos, H., Kaski, S.: Information retrieval perspective to nonlinear dimensionality reduction for data visualization. Journal of Machine Learning Research, Vol. 11, pp. 451–490 (2010)
45. Chang, C., Lin, C.: LIBSVM: A library for Support Vector Machines. ACM Trans Intell Syst Technol, Vol. 2, No. 3, p. 27 (2011)
46. Cortes, C., Vapnik, V.N.: Support vector networks. Mach Learn, Vol. 20, No. 3, pp. 273–97 (1995)

Una propuesta genetica y bayesiana para resolver problemas de clasificación en aplicaciones médicas

Alfredo Reyes M., Abraham Sánchez L., Enrique Mote R.

Benemérita Universidad Autónoma de Puebla,
Computer Science Department,

reyes-fred@hotmail.com, asanchez@cs.buap.mx,
mote.enrique.iti@gmail.com

Resumen. Diversos problemas del mundo real pueden ser modelados a través de las redes bayesianas. Este trabajo presenta el desarrollo de una red bayesiana clasificadora para aplicaciones médicas, dicha propuesta es un algoritmo hibrido combinado con algoritmos genéticos. En este artículo utilizamos una base de datos de 267 conjuntos de imágenes SPECT descargadas de SPECT Heart Data Set. Estos datos se procesan, utilizando dos algoritmos: K2 y nuestra propuesta genética-bayesiana. Una vez hecho esto, se determina si el paciente es catalogado como normal o anormal. Finalmente estos resultados son corroborados con el uso del software Netica para constatar el porcentaje de error que estos presentan y como se comportan cada uno de los algoritmos.

Palabras clave: Aplicaciones médicas, redes bayesianas.

1. Introducción

En [1] se presenta una propuesta para la evaluación de las redes bayesianas a partir de datos médicos. Los autores utilizan una cantidad importante de algoritmos de clasificación propuestos en la literatura, incluidos los de naturaleza bayesiana. Según los autores, los resultados obtenidos dependen tanto del proceso de clasificación de las redes bayesianas, como de la experiencia del médico, quien es el que lleva a cabo el proceso de captura e interpretación de los padecimientos presentados.

En [2], los autores proponen la aplicación de redes bayesianas en el modelado de sistemas expertos de triaje en los servicios de urgencias médicas. La propuesta del trabajo se realiza mediante la clasificación de datos provenientes de experiencias de triaje y de la opinión de médicos expertos.

El corazón humano es un sistema complejo ya que que brinda una diversa cantidad de características para el análisis de su estado y funcionamiento. Dentro de un diagnóstico de tomografías computarizadas de la emisión de protones únicos cardiacos (SPECT), algunos radionucleidos emisores de gamma se inyectan primero en el torrente sanguíneo de un paciente. Posteriormente, una

cámara gamma plana se hace girar alrededor del paciente para adquirir múltiples proyecciones 2D. La distribución 3D de una fuente de radionucleidos se puede construir a partir de las proyecciones 2D. Uno de los principales inconvenientes de la SPECT es su tiempo de formación de imágenes dinámicas. Una red bayesiana permite la obtención de una red clasificadora de nodos [3]. Es posible generar estos nodos a partir de los atributos que posee un data set, el cual es resultado del registro y tratamiento de datos correspondientes a mediciones tomadas de tomografías. Los resultados de la generación de una red bayesiana a partir de un algoritmo son muy diferentes, estos dependen de la clasificación y del ordenamiento que se lleva a cabo. Las redes bayesianas en este trabajo nos ayudarán a determinar si un paciente es normal o anormal, siempre y cuando cumpla cierto porcentaje en cada una de las características.

La generación de éstas redes y el procesamiento que tienen una vez generadas es diferente, puesto que la clasificación de nodos, el orden y la conexión entre ellos es diferente. A lo largo de este trabajo se utilizan dos algoritmos principales, el algoritmo K2 y un algoritmo genético hibrido, con estas dos propuestas se obtendrá una red bayesiana [4]. Además se utiliza el software Netica para observar gráficamente la red que se construye a partir de los datos. El desarrollo de estos algoritmos fue realizado en Matlab. Se presentan en la sección de resultados, los experimentos realizados que comprueban la efectividad de nuestra propuesta.

2. Redes bayesianas

Una red bayesiana, o red de creencia, es un modelo probabilístico multivariado que relaciona un conjunto de variables aleatorias mediante un grafo dirigido que indica explícitamente una influencia causal. Gracias a su motor de actualización de probabilidades, el Teorema de Bayes, las redes bayesianas son una herramienta extremadamente útil en la estimación de probabilidades ante nuevas evidencias. Una red bayesiana es un tipo de red causal.

Las redes bayesianas son un formalismo basado en la teoría de probabilidades y los grafos.

Una red bayesiana está definida por:

- un grafo orientado sin circuito $G = (V, E)$, donde V es el conjunto de nodos de G y E el conjunto de los arcos de G,
- un espacio de probabilidad finito (Ω, Z, p)
- un conjunto de variables aleatorias asociadas a los nodos del grafo y definido en (Ω, Z, p), tal que:

$$p(V_1, V_2, ..., V_n) = \prod_{i=1}^{n} p(V_i | C(V_i)) \tag{1}$$

donde $C(V_i)$ es el conjunto de las causas (padres) de V_i en el grafo G.

Una red bayesiana es por lo tanto un grafo causal que ha sido asociado con un representación probabilística subyacente. Como sabemos, esta representación

permite proporcionar un caracter cuantitativo a los razonamientos sobre la causalidad que se pueden hacer en el grafo.

Normalmente las redes bayesianas consideran variables discretas o nominales, por lo que si no lo son, hay que discretizarlas antes de construir el modelo. Los métodos de discretización se dividen en dos tipos principales (i) no supervisados y (ii) supervisados. Los métodos no supervisados no consideran la variables clase, así que los atributos continuos son discretizados independientemente. El método más simple es dividir el rango de valores de cada atributo, $[Xmin, Xmax]$, en k intervalos, donde k está dado por el usuario o se obtiene usando una cierta medida de información sobre los valores de los atributos. Los métodos supervisados consideran la variables clase, es decir los puntos de división para formar rangos en cada atributo y estos son seleccionados en función del valor de la clase. El problema de encontrar el número óptimo de intervalos y de los límites correspondientes se puede considerar como un problema de búsqueda. Es decir, podemos generar todos los puntos posibles de división para formar intervalos sobre la gama de valores de cada atributo, y estimamos el error de clasificación para cada partición posible [5].

3. Procesamiento de señales

La base de datos de 267 conjuntos de imágenes SPECT se procesa para extraer características que resumen las imágenes SPECT originales. Estos conjuntos de datos fueron descargados de SPECT Heart Data Set (disponible en https://archive.ics.uci.edu).

Este conjunto de datos nos describe el diagnóstico de tomografías computarizadas de la emisión de protones únicos cardiacos. Cada uno de los pacientes se clasifican en dos categorías: normales y anormales. Como resultado, se han creado 44 patrones de operación continuos para cada paciente. El patrón se procesa adicionalmente para obtener 22 patrones de funciones binarias. El algoritmo clip3 se utiliza para generar reglas de clasificación de estos patrones [6].

- Número de instancias: 267
- Número de atributos: 23 (22 binarios + 1 clasificación binaria)

El repositorio pone a nuestra disposición 3 archivos los cuales contienen una distribución de clases distintas.

Datos enteros

Clase	Número de ejemplo
0	55
1	212

Datos entrenados

Clase	Número de ejemplo
0	40
1	40

Datos de prueba

Clase	Número de ejemplo
0	15
1	172

4. Algoritmos

4.1. Algoritmo K2

El algoritmo K2 está basado en la optimización de una medida. Esa medida se usa para explorar, mediante un algoritmo de ascenso de colinas, el espacio de búsqueda formado por todas las redes que contienen las variables de la base de datos. Se parte de una red inicial y esta se va modificando (añadiendo arcos, borrándolos o cambiándolos de dirección) obteniendo una nueva red con mejor medida. En concreto la medida de K2 para una red G y una base de datos D es la siguiente [7]:

$$f(G:D) = \log P(G) + \sum_{i=1}^{n} \left[\sum_{k=1}^{si} \left[\log \frac{\Gamma(n_{ik})}{\Gamma(N_{ik}, n_{ik})} + \sum_{j=1}^{ri} \log \frac{\Gamma(N_{ik}, n_{ik})}{n_{ik}} \right] \right] \quad (2)$$

donde:

- N_{ik} es la frecuencia de las configuraciones encontradas en la base de datos D de las variables x_i.
- n es el número de variables, tomando su j-ésimo valor y sus padres en G tomando su k-ésima configuración,
- si es el número de configuraciones posibles del conjunto de padres,
- Γ_i es el número de valores que puede tomar la variable x_i,
- $N_{ik} = \sum_{j=1}^{\Gamma_i} N_{ik}$ y Γ es la función Gamma.

4.2. Algoritmo genético

Los algoritmos genéticos (AGs) son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización [9]. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin.

Los algoritmos genéticos usan una analogía directa con el comportamiento natural. Trabajan con una población de individuos, cada uno de los cuales representa una solución factible a un problema dado. A cada individuo se le asigna un valor o puntuación, relacionado con la bondad de dicha solución. En la naturaleza esto equivaldría al grado de efectividad de un organismo para competir por unos determinados recursos.

Una generación se obtiene a partir de la anterior por medio de los operadores de reproducción. Existen dos tipos: Cruza: Se trata de una reproducción de tipo

sexual. Se genera una descendencia a partir del mismo nuúmero de individuos de la generación anterior. Existen varios tipos que no se detallarán en este trabajo. Copia: Se trata de una reproducción de tipo asexual. Un determinado número de individuos pasa sin sufrir ninguna variación directamente a la siguiente generación. Si desea optarse por una estrategia elitista, los mejores individuos de cada generación se copian siempre en la población temporal, para evitar su pérdida. A continuación comienza a generarse la nueva población en base a la aplicación de los operadores genéticos de cruza y/o copia. Una vez generados los nuevos individuos se realiza la mutación con una probabilidad Pm. La probabilidad de mutación suele ser muy baja, por lo general entre el 0,5 % y el 2 %. Se sale de este proceso cuando se alcanza alguno de los criterios de parada, establecidos en el problema a resolver.

Para la aplicación dentro del problema de la generación de estructuras DAG (grafo acíclico dirigido) para redes bayesianas, hemos desarrollado un algoritmo genético que funciona en colaboración con el software Netica, el cual contiene una API de desarrollo útil para la evaluación de resultados de implementaciones de redes bayesianas y la evaluación de las estructuras de las mismas.

El algoritmo que usamos se basa en la descripción que se ha dado anteriormente, se inicia con un DAG aleatorio, el cual es nuestra población inicial, seguido de este paso, se debe evaluar el rango de error de los nodos usando la API del software Netica para obtener un parámetro comparable con las siguientes generaciones que se puedan obtener. Este parámetro se almacena y la siguiente generación se genera para su evaluación, lo que nos otorga un nuevo DAG que debe ser evaluado nuevamente con la API. Hemos establecido un criterio de permanencia del parámetro óptimo (rango de error de los nodos) para poder detener las iteraciones del algoritmo genético cuando se ha encontrado una solución factible, este criterio indica que el rango del DAG almacenado debe prevalecer durante 30 iteraciones para poder ser considerado como un DAG factible para formar la red bayesiana correspondiente a los datos en el dataset. Estableciendo el algoritmo de manera general, quedaríde la siguiente manera.

1. Generación del DAG inicial (aleatorio)
    a) Evaluación del rango de error del DAG
    b) Almacenamiento de rango de error y el DAG
    c) Aumentar número de iteraciones en 1
2. Repetir hasta que el número de iteraciones llegue a 30
    a) Generar nuevo DAG (población)
    b) Evaluar rango de error del nuevo DAG
    c) Si rango de error del nuevo DAG es menor al rango de error del DAG almacenado
        i) Almacenar el nuevo DAG para comparar
        ii) Reiniciar conteo de iteraciones
    d) sino
        i) Aumentar conteo de iteraciones
3. Si el número de iteraciones llega a 30
    i) Devolver el DAG almacenado

5. Resultados experimentales

Se realizaron distintas pruebas para obtener distintos resultados con el uso del algoritmo K2. El algoritmo fue implementado con las libreras existentes de MATLAB. Los entrenamientos de la estructura de la red se llevaron en 2 pruebas distintas con el mismo data set; se tomaron el data set de entrenamiento (80 casos) y el data set de prueba (187 casos). Además, se usaron solamente las primeras 14 características, más el atributo objetivo. El orden de entrada de los nodos para ambos casos fue el siguiente: 10 14 9 4 2 8 12 1 3 5 7 13 11 15 6.

A continuacin se muestran imágenes de las topologías de red bayesiana que se obtuvieron.

```
DAG =

     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     1     0     0     0     0     1     0     0     0     0
     0     0     0     0     0     0     1     0     0     0     0     0     1     0     0
     0     1     0     0     1     0     0     1     0     0     0     0     0     0     0
     0     0     0     0     0     1     0     0     0     0     0     0     0     0     1
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     1     0     1     0     0     0     0     0     0     0     0     1     1     0     0
     1     0     0     1     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     1     0     0     0     1     0     0     0     0     1     1
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     1     0     1     0     0     0     1     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     1     0     0     1     0     0     0     0     1     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
```

Fig. 1. DAG en forma de matriz para el dataset de 187 casos.

El siguiente paso es verificar el margen de error de las redes obtenidas mediante el software Netica.

Lo siguiente es entrenar la red y verificar su rango de error. Una vez hecho esto, obtenemos:

Para la parte de comparación, obtuvimos los siguientes datos del algoritmo genético: Con un total de 25 ejecuciones para el algoritmo, pudimos observar que la configuración para la red bayesiana del data set fue la siguiente, ver la figura 3.

El algoritmo genético nos proporcionó una matriz binaria, la cual pudimos mapear a la red bayesiana presentada en la figura 3. Al entrenar la red con los casos del data set, pudimos observar los resultados de los rangos de error para cada nodo de la red; estos datos se presentan en la tabla siguiente.

Se presenta a continuación una gráfica en la cual se puede observar la diferencia en el rango de error para cada nodo de la red bayesiana, para el algoritmo K2 y para el algoritmo genético.

El rango de error obtenido para cada nodo de la red bayesiana en el algoritmo genético indica el porcentaje de equivocación que tiene cada nodo de la red

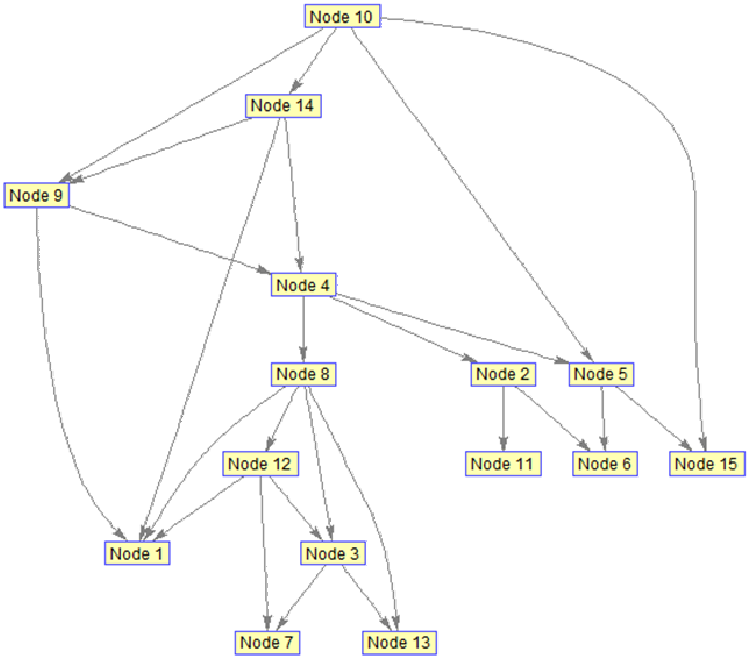


Fig. 2. Red bayesiana resultante.

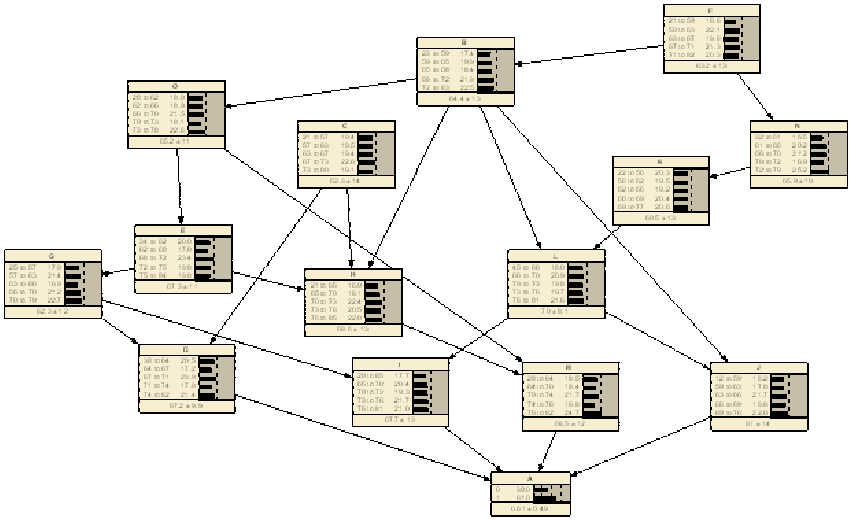


Fig. 3. Red bayesiana obtenida del DAG.

Tabla 1. Rango de error de los nodos para el DAG obtenido del algoritmo K2.

Nodo	Rango de error
A	8.021 %
B	80.21 %
C	79.68 %
D	75.4 %
E	77.54 %
F	80.75 %
G	77.54 %
H	77.54 %
I	77.54 %
J	80.21 %
K	78.07 %
L	78.07 %
M	66.84 %
N	80.75 %
O	73.8 %

Tabla 2. Rango de error de los nodos para el DAG obtenido del algoritmo genético.

Nodo	Rango de error
A	7.034 %
B	76.47 %
C	77.01 %
D	75.4 %
E	78.07 %
F	81.28 %
G	77.54 %
H	79.14 %
I	79.14 %
J	75.94 %
K	79.86 %
L	76.47 %
M	75.4 %
N	72.73 %
O	77.54 %

en base a la evaluación del data set y de la forma en la que los nodos están relacionados.

Es posible observar que dentro de estas algoritmos de clasificación se presentan resultados muy semejantes, sin una variación muy grande. Llegamos a la conclusión de determinar que el algoritmo genético nos brinda mejores resultados, en base a los porcentajes de error que Netica nos permite conocer a través de su software. En estos porcentajes se puede observa una pequeña variación, aunque cabe resaltar que el algoritmo K2 en ciertos nodos el porcentaje es menor al genético. La propuesta del algoritmo genético aún es muy reciente, por lo que existen diversas variaciones que se pueden agregar a este para obtener

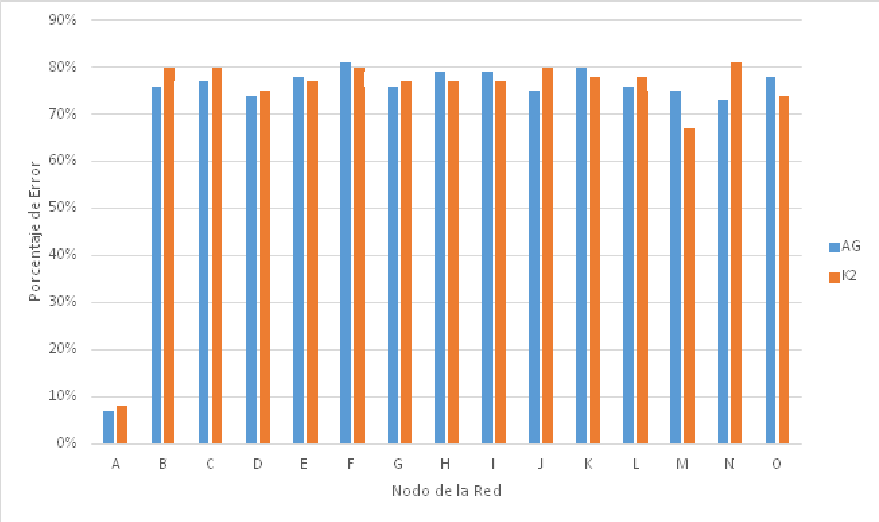


Fig. 4. Gráfica comparativa entre los algoritmos K2 y genético.

una mejor evaluación de las instancia.

Los tiempos de ejecución no se pueden determinar como una medida cualitativa para reportar puesto que estos fueron muy pequeños para ambos casos. En la corrida de los algoritmos para esta base de datos fue de apenas unos segundos, sin una variación realmente significativa, que nos permitiría dar una respuesta favorable a cierto algoritmo.

6. Conclusiones y trabajo futuro

Los resultados que obtuvimos para la estructura de la red bayesiana en el algoritmo K2 y el algoritmo genético (AG) muestran discrepancias, sin embargo, el porcentaje de error que se obtiene en los nodos de la red del AG es menor en el mayoría de los casos, lo que indica que la estructura es más favorable para el diagnóstico del nodo objetivo (A). Para cada uno de los algoritmos, se exige un trabajo mayor, debido a la complejidad de cada uno, ya que, por ejemplo, el algoritmo K2 exige una introducción de los nodos de manera ordenada, sin embargo, este orden debe ser generado de manera óptima para poder asegurar una estructura de la red más efectiva.

Por otro lado, el algoritmo genético exige un mayor número de ejecuciones para poder otorgar una estructura más confiable, ya que pueden darse casos de atasco en mínimos o máximos locales.

El trabajo a futuro de esta propuesta consiste en implementar los métodos de optimización de parámetros para ambos algoritmos, en cuestiones de orden y número de ejecuciones. También se tiene como objetivo futuro extender el rango

de aplicación del data set para poder cambiar el objetivo de los nodos y el sentido de la red.

Referencias

1. Barrientos, M. R., Cruz, R. N., Acosta, M. H. G., Rabatte, S. I., Pavón L. P., Gogeascoechea, T. M. C., Blázquez, M. M. S.: Evaluación del potencial de redes bayesianas en la clasificación en datos médicos. Revista Médica de la Universidad Veracruzana, Vol. 8, No. 1 (2008)
2. Abad-Grau, M. M., Ierache, J. S., Cervino, C.: Aplicación de redes bayesianas en el modelado de un sistema experto de triaje en servicios de urgencias médicas. IX Workshop de Investigadores en Ciencias de la Computación, Argentina, pp. 43–47 (2007)
3. Neapolitan, R.E.: Learning Bayesian networks. Pearson - Prentice Hall (2003)
4. Heckerman, D.: A tutorial on learning with Bayesian networks. Microsoft Research (1995)
5. Sierra Araujo, B. (Coordinador): Aprendizaje automático: Conceptos basicos y avanzados. Capítulo 6, Pearson, Prentice-Hall (2006)
6. Cios, K. J., Wedding, D. K., Liu, N.: CLIP3: cover learning using integer programming. Kybernetes, Vol. 26, No. 4-5, pp. 513–536 (1997)
7. Cooper, G. F., Herskovits, E.: A Bayesian method for the induction of probabilistic networks from data. Machine Learning, Vol. 9, pp. 309–348 (1992)
8. Myers, J., Laskey, K. B.: Learning Bayesian networks from incomplete data with stochastic search algorithms. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), pp. 458–465 (1999)
9. Haupt, R. L., Haupt, S.E.: Practical genetic algorithms. Wiley-Interscience, 2nd Edition (2004)
10. Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., et al.: Bayesian data analysis. CRC Texts in Statistical Science, Third Edition (2013)

Propagación de malware: propuesta de modelo para simulación y análisis

Luis Angel García Reyes[1], Asdrúbal López-Chau[1], Rafael Rojas Hernández[1], Pedro Guevara López[2]

[1] Universidad Autónoma del Estado de México, CU UAEM Zumpango, Zumpango, Estado de México, México

[2] Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Culhuacán, Ciudad de México, México

angel_garc@outlook.es, alchau@uaemex.mx, http://www.alchau.com

Resumen. La cantidad y diversidad de software malicioso (Malware) en la actualidad es enorme. Se espera que en un corto tiempo se incrementen todavía más las amenazas cibernéticas. Uno de los enfoques más utilizados para enfrentar este problema, es analizar la dinámica de propagación de Malware utilizando modelos matemáticos basados en sistemas de ecuaciones diferenciales propuestos en la década de los años 20. Una desventaja de ese enfoque es la dificultad que se presenta para relacionar el valor de los parámetros de las ecuaciones con aspectos específicos del mundo real. En este artículo, se presenta una propuesta para simular y analizar la propagación de Malware considerando elementos que se presentan cotidianamente en la realidad. Se propone un modelo en el que se introducen dos conceptos nuevos, el primero es remarcar la diferencia entre individuos y dispositivos; el segundo es realizar la distinción entre propietarios y usuarios de dispositivos. Estos elementos son introducidos como parámetros del modelo para analizar la evolución de la propagación. El modelo presentado se implementó como una plataforma funcional, que se utilizó para realizar simulaciones con 1,000 dispositivos. De acuerdo a los resultados de los experimentos realizados, se encuentra evidencia del efecto de los conceptos introducidos sobre la dinámica de propagación de Malware.

Palabras clave: Malware, propagación, plataforma de simulación, virus.

Malware Propagation: Proposal of the Model for Simulation and Analysis

Abstract. The amount and diversity of malicious software (Malware) today is huge. It is expected a further increase of cyberthreat in a short

time. One of the most commonly used approaches to address this problem is to analyze the dynamics of Malware propagation using mathematical models based on systems of differential equations. These were proposed in the decade of the 20s. One disadvantage of this approach is the difficulty presented to relate the value of the parameters of the equations with specific aspects of the real-world. In this paper, a proposal is presented to simulate and analyze the spread of Malware considering elements that occur daily in reality. A model in which two new concepts are introduced is proposed, the first concept is to emphasize the difference between individuals and devices; the second one is to make the distinction between owners and users of devices. These elements are introduced as parameters of the model to analyze the evolution of the spread. The model was implemented as a functional platform, which was used for simulations with 1,000 devices. According to the results of experiments, it is evidence of the effect of the concepts introduced on the dynamics of spreading Malware.

Keywords: Malware, propagation, simulation platform, virus.

1. Introducción

La seguridad de los sistemas informáticos es considerada como un aspecto vital tanto por empresas como por usuarios finales. Uno de los elementos más importantes que ponen en peligro la seguridad de este tipo de sistemas es la presencia de programas malignos o software malicioso (Malware) en el sistema operativo (SO) de los dispositivos.

De acuerdo a los últimos reportes de algunas de las principales empresas dedicadas a la seguridad informática [10], [11], [5], la tendencia en los próximos meses es que aumente tanto el número como las variantes de las amenazas.

Con el objetivo de enfrentar al Malware, empresas e investigadores han invertido esfuerzos considerables para tratar de entenderlo desde diversos ángulos. Uno de los aspectos que ha atraído la atención de la comunidad científica en los últimos años, es la creación de modelos de la propagación del software malintencionado. Esto debido a que con ellos se pueden realizar estimaciones sobre la rapidez con la que un Malware podría infectar a todos los dispositivos en una red, o incluso a una cantidad considerable de dispositivos del mundo entero, lo que permite tomar medidas antes de que ocurran daños significativos en las organizaciones.

La mayoría de los modelos de propagación de Malware propuestos hasta la fecha, están basados en los modelos deterministas epidemiológicos tipo SIR (Susceptible, Infectado y Recuperado), y una plétora de variantes. Estos modelos usan ecuaciones diferenciales de primer orden con coeficientes constantes. La principal ventaja de estos modelos es su simplicidad. Sin embargo, uno de los problemas es que para la mayoría de las aplicaciones, no resulta sencillo relacionar el valor de los coeficientes con elementos del mundo real.

En este artículo, presentamos una plataforma para estudiar la propagación de Malware implementado como una aplicación de escritorio Java (Para obtener el código fuente ver [4]). La plataforma presentada contiene una serie de parámetros simples de entender, que están directamente relacionados con elementos reales, tal como la cantidad de usuarios, cantidad de dispositivos, número de dispositivos infectados al inicio de la simulación, y otros. La aplicación permite simular paso a paso la dinámica de la infección, y presentar los resultados de manera tabular y gráfica.

El resto del artículo se encuentra organizado en 6 secciones. La Sección 2 presenta la definición Malware, así como los principales tipos y formas de propagación más conocidas. Una revisión de los trabajos relacionados se encuentra en la sección 3. El modelo propuesto y su implementación son presentados en la sección 4. Los resultados de las simulaciones y una discusión de los mismos están presentes en la sección 5. Las conclusiones de la investigación y trabajos futuros pueden leerse en la sección 6.

2. Malware

El término Malware proviene del inglés "malicious software", que en español significa código malicioso [3], [6]. Técnicamente, Malware se refiere a programas que se instalan en los SO de los equipos con desconocimiento de los usuarios. Estos programas esperan silenciosamente su ejecución con la intención de causar daños con acciones inadecuadas u objetivos maliciosos. Por lo tanto, hablar de software malicioso involucra amenazas constantes en los SO.

Existen diferentes tipos de software malicioso. Un virus informático infecta los dispositivos viajando de manera autónoma entre ellos, normalmente, esperando a ser detonado por un usuario final. Un gusano (Worm) es programado también para viajar entre los dispositivos, pero este tipo de software sólo se instala una vez dentro del sistema, y posteriormente busca otro dispositivo para su infección. Algunos gusanos requieren de la interacción con usuarios, pero también existen algunos de ellos que logran infectar sin la necesidad de dicha interacción. Los troyanos (Trojans), por otro lado, hacen honor a la leyenda mítica griega "Caballo de Troya", debido a que el software no aparenta ser mal intencionado, sino todo lo contrario, parece ser un software útil para el usuario. Los troyanos también pueden ser instalados sin la necesidad de ser descubiertos o detonados por un usuario, permitiéndose así el acceso al sistema sin aviso alguno. Estos son mejor conocidos como troyanos de puerta trasera (Trojans Backdoors). A diferencia de los virus y los gusanos, los troyanos dependen del acceso a Internet. Otro tipo de Malware que se ha popularizado en los últimos años, son los llamados Spyware del tipo de infiltración silenciosa. Estos pretenden la obtención de información de los usuarios. Entre los datos más importantes que puede llegar a sustraer esta amenaza, se encuentran las contraseñas y números de tarjetas de crédito.

Un tipo de amenaza llamado *phishing* ha tenido una gran actividad recientemente. La finalidad del phishing es sustraer información de usuarios

usando una estrategia diferente a los troyanos. Se presenta a los usuarios una interfaz (página Web) de alguna entidad de confianza, por ejemplo banco u otra organización empresarial. El usuario puede ser engañado y proporcionar datos tales como su nombre de usuario, contraseña o número generado por dispositivo electrónico "token". La forma de hacer llegar a los usuarios los enlaces es a través de correo electrónicos o mensajes [10].

Actualmente, diversos tipos de Malware son capaces de burlar a los sistemas de prevención implementados en diversos SO, esto debido principalmente a la falta de mantenimiento o de actualización de los equipos con acceso a Internet considerando aspectos cómo la falta de cultura informática por parte de los usuarios, lo que les impide tomar las medidas necesarias contra este tipo de software.

Hay diferentes maneras de infectar los dispositivos con Malware. Una cantidad importante de ellas se relaciona con las actividades que se realizan cotidianamente, como la recepción de correo electrónico y mensajes con archivos adjuntos contaminados. Otra forma es visitando páginas Web aparentemente inofensivas, pero que tienen vínculos a descargas y/o instalación Malware. El compartir archivos en red o a través de algún medio extraíble como lo son CDs, USB, HDD portable, DVD entre otros, es también otra manera de infectar equipos con Malware [1].

Por otro lado, las técnicas de propagación de estos programas maliciosos los hacen capaces de multiplicarse bajo la intrusión a través de la red e infectar a un sinnúmero de sistemas [3], [6].

3. Trabajos relacionados

Uno de los enfoques más ampliamente utilizados para modelar la propagación de Malware es la aplicación del modelo epidemiológico SIR (o simplemente SIR de aquí en adelante) desarrollado por Kermack y McKendrick en 1927 [7] para estudiar la propagación de enfermedades en cortos periodos de tiempo. SIR permite estimar la cantidad de individuos de una población que son susceptibles de contraer una enfermedad, así como la cantidad de individuos de la población que han sido infectados por esa misma enfermedad y el número de individuos que se han recuperado.

Las ecuaciones diferenciales ordinarias que modelan la dinámica de infección son las siguientes [8]:

$$\frac{dS(t)}{dt} = -\beta I(t)S(t) \qquad \frac{dI(t)}{dt} = \beta I(t)S(t) - \alpha I(t),$$
$$\frac{dR(t)}{dt} = \alpha I(t),$$

cuya solución es:

$$S(t) = S(0)e^{\frac{-\beta}{\alpha}R},$$
$$I_{max} = -\frac{\alpha}{\beta} + \frac{\alpha}{\beta}\ln\left(\frac{\alpha}{\beta}\right) + S(0) + I(0) - \frac{\alpha}{\beta}\ln S(0),$$

donde:

- Se asume que una cantidad de individuos susceptibles S(t), se encuentra en contacto con individuos ya infectados I(t), a través de una mezcla homogénea. Además, cada individuo es idéntico al resto.
- La cantidad de individuos que se recuperan (y que no pueden volver a infectarse) se representa con R(t).
- El tamaño N de la población es constante, lo que implica que N=S(t)+I(t)+R(t).
- El número de individuos infectados que contagia a otros susceptibles a una tasa de infección β (denominada *transmission rate constant*). Cada individuo infectado es infeccioso.
- La tasa de recuperación de los individuos es α. Si no se considera la recuperación, entonces este parámetro es igual a cero.
- I_{max} es el máximo número de individuos infectados en la epidemia.

Desde la aparición de SIR, se han desarrollado variantes interesantes aplicadas a diversas topologías de redes de dispositivos electrónicos.

En [9], se muestra un estudio detallado de la aplicación de un modelo analítico para el proceso SIS (por las siglas en inglés *susceptible infected susceptible*). En este modelo, se usa el concepto de çontactos", que son usados para propagar un virus informático o biológico. La probabilidad de que un individuo se infecte está en función del promedio de los vecinos infectados. Aunque analíticamente este modelo es atractivo, la dificultad más notable es la complejidad para relacionar directamente sus parámetros con elementos del mundo real.

Un enfoque diferente para estudiar la propagación de Malware, es emplear una simulación por computadora. En [2] se propone EpiNet, un framework para simular propagación de gusanos informáticos en redes masivas Bluetooth de smartphones. Tanto el modelo propuesto, como las conclusiones presentadas en [2], son interesantes e importantes para entender el proceso de propagación de gusanos informáticos. Sin embargo, entre las desventajas más importantes de este framework se pueden mencionar las siguientes: 1) se considera que dos dispositivos conectan si están lo suficientemente cercanos físicamente (10 m, para dispositivos BT clase II). 2) Además, la topología de la red toma un rol importante en EpiNet. Ambas consideraciones no son adecuadas actualmente, debido a restricciones tales como el ahorro de energía (Bluetooth apagado), o la necesidad de que los usuarios autoricen las conexiones a redes.

En la siguiente sección, se presenta el modelo propuesto en este artículo para analizar la propagación de Malware.

4. Modelo propuesto

En el diseño del modelo presentado, se tomaron en cuenta algunos aspectos del mundo real que se consideran importantes, y que hasta donde sabemos, no se encuentran presentes en otros artículos publicados anteriormente en la literatura especializada.

El primer aspecto es la diferencia entre dispositivo e individuo. Los dispositivos pueden infectarse, mientras que los individuos no. Por lo tanto, a diferencia de otros modelos, en el nuestro, un mismo individuo puede a veces contagiar a otro dispositivo, dependiendo si usa un equipo infectado o no para enviar mensajes.

El segundo aspecto es la introducción del concepto de usuario y propietario. Un usuario puede usar equipos diferentes para enviar o recibir mensajes, mientras que un propietario siempre usa el mismo dispositivo. Estos conceptos son importantes, ya que permiten capturar la realidad que ocurre en escuelas, cafés Internet o cualquier otro espacio donde varios usuarios comparten equipos.

4.1. Elementos del modelo propuesto

Los principales elementos del modelo propuesto son los siguientes:

1. *Dispositivo.* Se refiere a una computadora, teléfono inteligente o cualquier otro equipo electrónico capaz de enviar y recibir mensajes de diversos tipos, y que pueda contagiarse de Malware.
2. *Mensaje.* Los mensajes pueden ser correo electrónico, enlaces, texto o multimedia, enviados o recibidos por aplicaciones que se ejecutan en dispositivos electrónicos. La infección se da al leer desde un dispositivo susceptible, un mensaje que ha sido enviado desde un dispositivo infectado.
3. *Usuario.* Son individuos que usan dispositivos electrónicos para recepción y envío de mensajes. En nuestro modelo, a diferencia de otros, los individuos pueden usar varios dispositivos infectados para propagar Malware.
4. *Propietario.* Son individuos que siempre usan el mismo dispositivo para la comunicación con otros individuos.
5. *Lista de contactos.* Cada individuo tiene una lista de contactos a los cuales envía mensajes. Este concepto es semejante al usado en [9].

El modelo del mecanismo de infección se propone simple, pero lo suficientemente flexible para poder adaptarse a mecanismos de infecciones más

avanzados, tales como robo de lista de contactos y envío automático de mensajes. Algoritmo 1 muestra el proceso general de infección implementado.

Algoritmo 1: Infección de dispositivos con Malware

Input: ;
M: Mensaje recibido;
U_i: individuo i-ésimo;
D_j: Dispositivo actual usado para leer M
Output: Nada

```
Individuo U_i lee mensaje recibido M desde dispositivo D_j
if M contiene Malware adjunto then
    D_j es infectado
    D_i adjuntará Malware la siguiente vez que sea usado para enviar
    mensaje.
```

La simulación de la propagación de Malware sigue el proceso mostrado en Algoritmo 2.

Algoritmo 2: Proceso general de la simulación

Input: ;
N: Número de individuos;
D: Número de dispositivos;
P: Número de propietarios;
TotalPasos: Número de pasos en la simulación;
S: Semilla del generador de números pseudoaleratorios;
C_{max}, C_{min}: Número de máximo y mínimo de contactos;
T_{max}: Número de contactos a los que se le envía mensaje
Output: Total dispositivos infectados

```
Crear lista de contactos aleatoriamente para cada individuo.
Asignar computadoras a usuarios
Asignar computadoras a propietarios
for paso=1: TotalPasos do
    foreach individuo con equipo asignado do
        Leer mensajes recibidos
        Enviar un mensaje a un máximo de T_max individuos de la lista de
        contactos, estos son elegidos pseudo-aleatoriamente.
    Re asignar computadoras a usuarios (propietarios se mantienen en el mismo
    dispositivo)
return Número de dispositivos infectados
```

Las principales características del modelo propuesto son las siguientes:

- Dispositivos vs individuos. A diferencia de los modelos epidemiológicos como SIR, donde un individuo infectado no puede volver a infectarse, en el modelo propuesto un individuo puede contribuir a infectar más de un dispositivo. Además, los individuos no se contagian.

- Independencia de la topología de red. Esto se consideró así, ya que actualmente, no es necesario estar conectado a una misma red para envío y recepción de mensajes.
- Mensaje como medio de infección. Estos mensajes pueden ser enlaces o Malware.
- Posibilidad de rotación entre dispositivos e individuos. Un mismo dispositivo puede ser usado por varios individuos (usuarios), o siempre por el mismo (propietario). La finalidad de esto es capturar la realidad de que los equipos se pueden compartir.

En la siguiente subsección, se presentan algunos detalles de la implementación del modelo desarrollado.

4.2. Plataforma desarrollada

La plataforma desarrollada fue programada en lenguaje Java y su arquitectura general se muestra en la Figura(1).

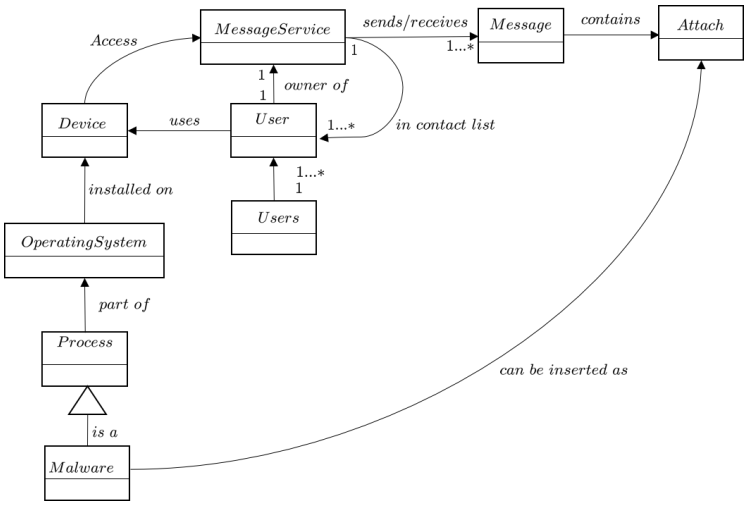


Fig. 1. Arquitectura general propuesta de la plataforma para simulación de propagación de Malware.

En la arquitectura propuesta se puede observar que cada *Individuo* no se contagia, sino un *Dispositivo* que es usado para leer *Mensajes*. El módulo Malware está relacionado con el módulo Dispositivo (Device en la Figura 1) a través del Sistema Operativo. Un dispositivo infecta a otro cuando le envía un anexo (Malware) en un mensaje, esto sin conocimiento del usuario del dispositivo infectado.

Aunque no es parte fundamental de la arquitectura general, en la implementación también se encuentran presentes otros elementos que permiten monitorizar el estado de los dispositivos, la asignación de dispositivos a usuarios y la cantidad de dispositivos infectados, entre otras variables.

5. Experimentos y resultados

En los modelos epidemiológicos usados para simular la propagación de Malware, puede modificarse el valor de unos cuantos parámetros, y observar la solución del sistema de ecuaciones diferenciales, presentando únicamente la cantidad de individuos infectados, susceptibles o recuperados por unidad de tiempo. Aunque esto es útil debido a su sencillez, no se permite explorar en detalle varios aspectos del mundo real que intervienen, como por ejemplo, cómo afecta a la velocidad de propagación la cantidad de mensajes recibidos o enviados, la relación entre la cantidad de usuarios en la lista de contactos y la tasa de propagación, etc.

En esta sección, se realiza una exploración de varios factores del mundo real que son capturados por el modelo propuesto, y se presentan las gráficas que muestran la evolución de la propagación. Se eligieron las siguientes variables para los experimentos:

1. Número de equipos infectados inicialmente. Esta simulación es común encontrarla en publicaciones similares, debido a que permite conocer en qué tiempo se espera tener a toda la población infectada, en función de los dispositivos inicialmente infectados de Malware.
2. Número de propietarios y número de dispositivos infectados inicialmente. El concepto de propietario y usuario es un aspecto novedoso en nuestro modelo, por lo que esta simulación no ha sido presentada anteriormente.
3. Tamaño de la lista de contactos. Este parámetro tiene que ver con la cantidad de usuarios que tienen contacto entre sí, y que por ende pueden contagiarse.

En experimentos preliminares, se encontró que la evolución de la propagación de Malware tiene una tendencia similar con diferentes números de dispositivos y de usuarios. Este comportamiento se encuentra presente también los modelos basados en ecuaciones diferenciales. Con la intensión de que en las gráficas presentadas se pueda apreciar mejor el comportamiento de la propagación de Malware, se decidió que el número de dispositivos y de usuarios fuera 1,000. Además, en cada experimento se realizaron 100 simulaciones, graficando el promedio de la cantidad de dispositivos infectados en cada unidad tiempo. Un número mayor de simulaciones, produce resultados con variación mínima en los resultados (menor al 0.05 %).

Usando los parámetros mencionados anteriormente, se encontró que la totalidad de equipos son infectados en menos de 50 pasos o unidades de tiempo, (parámetro TotalPasos en Algoritmo 2). En cada paso, el monitor implementado realizó un conteo de los equipos infectados. El generador de números pseudo-aleatorios de Java, usado para elegir individuos en la lista de

Cuadro 1. Parámetros usados en la simulación I y II

Parámetro	Sim I	Sim II
Total individuos (N)	1,000	1,000
Total dispositivos (D)	1,000	1,000
Propietarios (P)	15	5
Mínimo de contactos (C_{min})	5	5
Máximo de contactos (C_{max})	15	15
Contactos a los que se envía mensaje (T_{max})	1	1

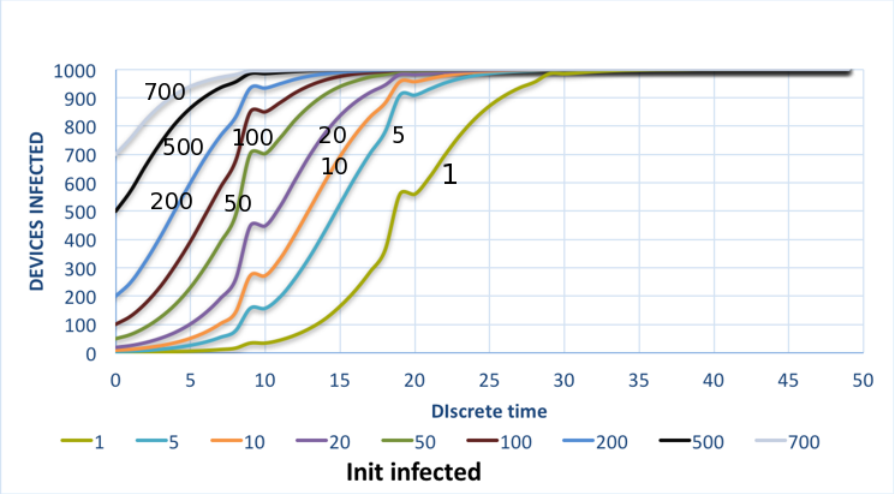


Fig. 2. Evolución de la propagación de Malware, simulación I.

contactos, toma como semilla el tiempo en que comienza un proceso de infección, usando el método estático *nanoTime()* de la clase System de Java.

Simulación I, número de dispositivos infectados inicialmente En este experimento, se varía la cantidad inicial de dispositivos que han sido infectados por Malware, mientras que los parámetros mostrados en la Tabla 1 se mantienen fijos.

La Figura 2 muestra la evolución de la propagación de Malware para esta simulación. Como es de esperarse, entre mayor sea la cantidad de dispositivos infectados inicialmente, el tiempo en que la totalidad de equipos se infecta es menor.

Es importante notar que en la Figura 2 parecería que hay una disminución en la cantidad de dispositivos infectados, sin embargo, esto no sucede en la versión actual nuestro modelo, ya que no se encuentra incorporado un mecanismo

de recuperación en el SO de los dispositivos simulados. Al analizar los datos, se encontró que esta aparente disminución se debe a un redondeo que hace la herramienta usada para graficar los datos.

Simulación II, número de propietarios y usuarios En el segundo experimento, se exploró cómo influye en la evolución de propagación de Malware, la cantidad de usuarios (propietarios) que siempre usan el mismo dispositivo para enviar o recibir mensajes . La cantidad de dispositivos infectados también se varió, con el objetivo de investigar si hay alguna efecto significativo en el comportamiento de la propagación. Los parámetros utilizados son los mostrados en la Tabla 1.

En la figura 3 puede observarse la cantidad de dispositivos infectados con respecto al tiempo. El parámetro P tiene un efecto interesante; la evolución de propagación de Malware pierde sensibilidad con respecto a la cantidad de equipos infectados inicialmente. Es decir, aún cuando la cantidad de dispositivos inicialmente infectados con Malware aumenta, el cambio principal en curva de crecimiento se concentra entre 10 y 15 unidades de tiempo.

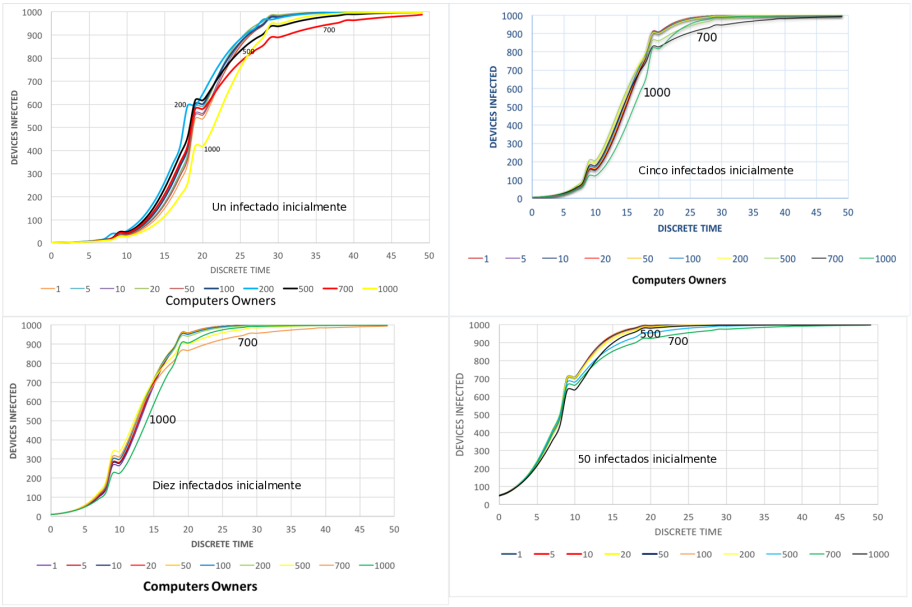


Fig. 3. Evolución de la propagación de Malware, simulación II.

Simulación III, tamaño de lista de contactos En el tercer experimento, se estudia el efecto que tiene el tamaño de la lista de contactos sobre la evolución de

la propagación de Malware. Los valores de los parámetros usados son similares a los de la simulación II, sólo se varían los valores del tamaño de la lista de contactos y el número de dispositivos infectados inicialmente . En la Figura 4 pueden observarse las gráficas resultantes.

En esta simulación, se observa que el número de dispositivos infectados al inicio, sí afecta notablemente el comportamiento de la propagación.

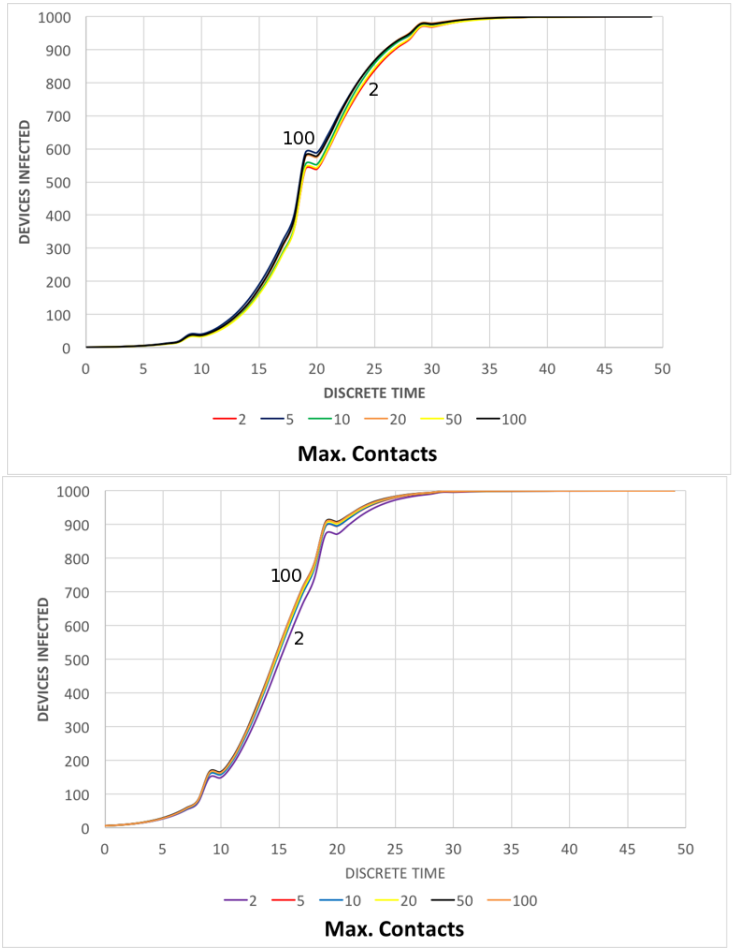


Fig. 4. Evolución de la propagación de Malware, simulación III.

6. Conclusiones y trabajo futuro

El estudio sobre propagación de Malware es de vital importancia actualmente, ya que el número de amenazas cibernéticas está en constante crecimiento. Uno de los enfoques para analizar la propagación de Malware, es utilizar representaciones matemáticas basados en el modelo SIR. En esta investigación, se propone un enfoque diferente para realizar este análisis. Se presenta el diseño y la implementación de una plataforma software para la simulación y el análisis de la propagación de Malware. El código fuente escrito en lenguaje Java puede ser descargado libremente usando la dirección mostrada en [4]. Esto con la finalidad de que otros investigadores puedan reutilizarlo.

La plataforma desarrollada está basada en un modelo que representa una abstracción de la realidad, en la que intervienen diversos elementos. Se presentan también dos aspectos novedosos; el primer aspecto que se propone es diferenciar entre usuario y dispositivo. De esta forma, los elementos que se infectan son los dispositivos y no los individuos, como ocurre en otros modelos. El segundo aspecto innovador es la introducción de los conceptos de usuario y propietario. Estos fueron considerados debido a que es común que un conjunto de equipos puede ser usado por varios individuos en diferentes tiempos, tal como ocurre en escuelas o cafés Internet.

Usando la plataforma propuesta, se puede analizar la dinámica de infección de una población con diferentes parámetros. En los experimentos presentados, se encontraron algunos efectos que tienen los parámetros en la evolución de la propagación de Malware.

Entre los trabajos futuros para esta investigación, se encuentran los siguientes: 1) realizar una comparativa de los resultados obtenidos con las respuestas que proporcionan los modelos epidemiológicos. Esto con el objetivo de encontrar la relación que existe entre los parámetros de las ecuaciones diferenciales, y aspectos del mundo real considerados en nuestra propuesta; 2) habilitar la plataforma para simulaciones de propagación de Malware a gran escala. Es decir, proveer la capacidad para simular millones de dispositivos e individuos; 3) implementar simulaciones de elementos de protección en los dispositivos, tales como programas anti Malware o anti SPAM y 4) Realizar más simulaciones para explorar la forma en que cada parámetro afecta a los otros.

Referencias

1. of Cambridge, U.: Computer viruses and other malware: what you need to know. http://www.ucs.cam.ac.uk/security/malware (2015)
2. Channakeshava, K., Chafekar, D., Bisset, K., Kumar, V.S.A., Marathe, M.: Epinet: A simulation framework to study the spread of malware in wireless networks. In: Proceedings of the 2Nd International Conference on Simulation Tools and Techniques. pp. 6:1–6:10. Simutools '09, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium (2009), `http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5652`

3. Fuentes, L.F.: Malware, una amenaza de internet. http://www.revista.unam.mx/vol.9/num4/art22/art22.pdf (Apr 2008)
4. García Reyes, L.A., López-Chau, A., Hernández, R., Guevara López, P.: Código fuente: Propagación de malware, propuesta de modelo para simulación y análisis. https://onedrive.live.com/redir?resid=993677954F59B48C!105155&authkey=!AOov_R3KxgRp1yM&ithint=folder % 2czip (May 2016)
5. Garnaeva, M., Wie, J.v.d., Makrushin, D., Ivanov, A., Namestnikov, Y.: Kaspersky security bulletin 2015. overall statistics for 2015. https://securelist.com/analysis/kaspersky-security-bulletin/73038/kaspersky-security-bulletin-2015-overall-statistics-for-2015/ (December 2015)
6. Jiménez Rojas, J.R., Soto Astorga, R.d.P.: Qué es malware? http://www.seguridad.unam.mx/usuario-casero/eduteca/main.dsc?id=193 (2009)
7. Kermack, W., McKendrick, A.: A contribution to the mathematical theory of epidemics. Proc. R. Soc. Lond. A 115 pp. 700,721 (1927)
8. Martcheva, M.: An introduction to mathematical epidemiology. Texts in applied mathematics, Springer, Boston, MA (2015), `http://cds.cern.ch/record/2112928`
9. Mieghem, P.V.: The viral conductance of a network. Computer Communications 35(12), 1494 – 1506 (2012), `http://www.sciencedirect.com/science/article/pii/S0140366412001405`
10. Security, P.: Pandalabs detected more than 21 million new threats during the second quarter of 2015, an increase of 432014. http://www.pandasecurity.com/mediacenter/news/pandalabs-detected-more-than-21-million-new-threats/ (September 2015)
11. Symantec: 2015 internet security threat report, volume 20. http://www.symantec.com/security_response/publications/threatreport.jsp (2015)

Arquitectura de integración para el desarrollo de un sistema de geo-recomendación para establecer puntos de venta

Edith Verdejo Palacios, Giner Alor Hernández, Cuauhtémoc Sánchez Ramírez, Susana Itzel Pérez Rodríguez, José Luis Sánchez Cervantes, Lisbeth Rodríguez Mazahua

Instituto Tecnológico de Orizaba, División de Estudios de Posgrado e Investigación, Orizaba, México

everdejo@acm.org, {galor, csanchez, lrodriguez }@itorizaba.edu.mx, jsanchezc@ito-depi.edu.mx,

Resumen. Los sistemas de geo-recomendación presentan la capacidad de realizar recomendaciones de lugares a partir de los intereses de los usuarios. Esta característica es útil en la competencia comercial ya que permite analizar mejor el estudio de localizaciones del mercado. Actualmente se considera que un factor clave para obtener el éxito comercial es la ubicación de los negocios; de manera que a medida que sean más cercanas con respecto a la localización de sus clientes, mayores serán los ingresos de las empresas. En este artículo se propone el diseño de una arquitectura de integración para el desarrollo de un sistema de geo-recomendación para el establecimiento de puntos de venta. La arquitectura se basa en un diseño de capas donde las funcionalidades de sus componentes e interrelaciones están distribuidas para un mejor mantenimiento y escalabilidad. Como prueba de contexto se presenta un caso de estudio que permite describir la arquitectura propuesta.

Palabras clave: Geolocalización, sistemas de información geográfica, sistemas de recomendación.

Integrational Architecture for Developing a Geo-recommender System for Establishing Points of Sale

Abstract. The geo-recommendation systems have the ability to carry out recommendations of places according to users interests. This feature is useful in commercial domains because it allows analyzing the study of potential markets locations. Nowadays, the business locations is considered a main factor to achieve the business success; so the profits can be increased if the business is more closer with respect to the location of its customers. This paper proposes the design of an integration architecture for developing a geo-recommender system to locate points of sale. The architecture is based on a layered design where the functionality of its components and relationships are distributed for better

maintenance and scalability. In order to validate our proposal, we present a case study describing the proposed architecture.

Keywords: Geolocation, geographic information system, recommender systems.

1. Introducción

Los sistemas de geo-recomendación son un nuevo punto de vista de los mecanismos de recomendación ya que son capaces de ofrecer recomendaciones tomando en cuenta las ubicaciones geográficas del usuario y de los lugares [1]. La principal atracción que ofrecen los sistemas de geo-recomendación reside en combinar sistemas de recomendación con información geográfica, donde el ámbito principal es el de actividades de ocio [1, 2]. Recientemente se identificó que las empresas deben ofrecer sus servicios a los clientes de manera rápida y oportuna debido a sus atareados estilos de vida; de manera que la cercanía de la ubicación de un negocio con respecto a la localización de sus clientes tiene un factor clave en el éxito de un negocio, ya que implica riesgos de imagen corporativa y financiera de la empresa [3]; y la distancia entre el domicilio del comprador y la ubicación del vendedor conlleva a un gasto adicional de transporte y tiempo. De tal forma que en los últimos años algunas empresas realizan estudios de mercado para evaluar a la competencia y su posicionamiento; intentando captar el negocio de sus contrarios investigando al consumidor y al entorno que le rodea, a través del diseño de mejores estrategias para captar un mayor número de clientes y solventar la preocupación del comportamiento de sus ingresos y egresos [4, 5]. Por otro lado, el empleo de los Sistemas de Información Geográfica (SIG) a través de la construcción de modelos geográficos con integración del entorno socioeconómico ha conformado un nuevo punto de vista para el estudio del mercado, el cual no ha sido explotado lo suficiente por las empresas. Pero estos estudios presentan la gran limitante de no extraer el conocimiento de las necesidades de las empresas para que estas logren explotarlo las veces que sea necesario con el fin de obtener u ofrecer recomendaciones de acuerdo a su comportamiento empresarial. En cambio, la utilización de sistemas de recomendación permite ofrecer recomendaciones afines al comportamiento de los usuarios a través de técnicas que permiten analizar conductas.

La combinación de los sistemas de recomendación y los sistemas de información geográfica permitiría desarrollar nuevos sistemas de información que ofrezcan recomendaciones a partir de características económicas y demográficas; por lo que este artículo plantea el desarrollo de una arquitectura de sistema de geo-recomendación que sirva como base para hacerle frente al problema de selección óptima que enfrentan las empresas para la ubicación de sus instalaciones.

La estructura de este artículo es como sigue: la sección 2 presenta el estado del arte referente a sistemas de geo-recomendación y enfoques basados en dinámica de sistemas. La sección 3 presenta la arquitectura de integración propuesta. La sección 4 presenta un caso de estudio como prueba de concepto del sistema de geo-recomendación. Finalmente, se presentan las conclusiones de este artículo, así como también el trabajo a futuro.

2. Estado del arte

A continuación se presenta la revisión del estado del arte sobre los trabajos relevantes que están relacionados directa o indirectamente con la selección de ubicación. Por lo que se decidió clasificar los artículos de acuerdo a los que utilizan mecanismos de recomendación y Sistemas de Información Geográfica (SIG).

2.1 Sistemas de recomendación en diferentes dominios

Colombo et al. [1] desarrollaron un sistema de recomendación móvil híbrido sensible al contexto, en donde se descartan las películas que no están siendo presentadas en las salas de cine y desechan los horarios de las películas a las que el usuario es probable que no asista basado en la distancia entre la posición origen y destino a partir de la identificación de las preferencias del usuario. Noguera et al. [2] discutieron que los cambios en el turismo electrónico requieren que los servicios proporcionen a los usuarios información relevante de acuerdo a sus contextos físicos actuales, teniendo en cuenta los gustos y preferencias. De tal forma que se propuso la implementación móvil sensible al contexto 3D de los restaurantes de la provincia de Jaén, España. Li et al. [6] propusieron un sistema de recomendación basado en los cupones de descuento con el fin de promover los productos pertenecientes a las plataformas en línea; a través de la construcción de árboles para categorizar los productos y un tratamiento de los datos para construir la red de usuarios y recoger datos de comportamiento. Batet et al. [7] desarrollaron un sistema de recomendación para dispositivos móviles basado en agentes; donde se ofrecen recomendaciones sobre actividades cercanas e interesantes para el usuario. Yu et al. [8] propusieron un sistema de inferencia basado en servicios de ubicación y conocimiento, el cual busca construir el conocimiento a través una aplicación móvil y mediante esta información el sistema puede ofrecer recomendaciones.

2.2 Sistemas de información geográfica aplicados en estudios ambientales y urbanos

Castro et al. [9] tomaron el modelo genérico de epidemiología para entender, modelar y analizar por medio de la dinámica de sistemas los factores críticos en la propagación de epidemias y la ejecución en un SIG con el propósito de visualizar e interpretar las fluctuaciones de sanos, infectados y recuperados. Corner et al. [10] plantearon un estudio para conocer los efectos de la deposición de los residuos de las granjas de peces. El cual utilizó una combinación de hojas de cálculo y un SIG por medio de un módulo de dispersión. Vairavamoorthy et al. [11] plantearon la falta de una herramienta capaz de predecir los riesgos por la intrusión de agua proveniente de alcantarillas, drenajes y zanjas a los sistemas de distribución de agua por medio del desarrollo un software predictor de riesgos asociados a los sistemas de distribución de agua fundamentados en SIG. Radiarta et al. [12] presentaron una evaluación multicriterio basada en un SIG que utiliza datos de detección satelital y datos de verificación de campo para identificar los sitios más adecuados para el desarrollo de la producción de vieira japonesa. Xu y Volker [13] formularon un estudio sobre las áreas residenciales en el desarrollo urbano, el cuál consistió en el análisis del SIG y de los

sistemas dinámicos (modelo y visualización 3D) y la visualización espacial en 2D. Por otra parte, Suarez et al. [14] indicaron que la competencia y el desempeño de las franquicias son afectados en parte por los factores de selección en la ubicación y la calidad de las instalaciones. De manera que se emplearon modelos de localización competitiva y herramientas SIG. De manera similar, Roig et al. [15] desarrollaron una metodología para el proceso de selección de puntos de venta, en donde se utiliza SIG para visualizar los datos espaciales que influyen en la toma de decisiones y al proceso de jerarquía analítica (AHP), el cual consiste en definir un modelo a través de los criterios asociados a la localización y las alternativas de ubicación mediante un análisis en la geodemanda y geocompetencia. Finalmente, Casillas et al. [16] propusieron un estudio para conocer la intensidad y la hora en que ocurre la isla urbana de calor, a través de la interpolación de temperaturas generadas en un SIG.

La Tabla 1 presenta un análisis comparativo de la literatura, tomando en cuenta los SIG, Sistemas de Recomendación y Dinámica de Sistemas.

Tabla 1. Análisis comparativo de la literatura.

NU=No Utilizado	**NE= No Especificado**	**H=Híbrido**	**BC=Basado en Conocimiento**	
Artículo	**Objetivo**	**Sistema de recomendación**	**SIG**	**Dinámica de Sistemas**
Castro et al. [9]	Plantear, desarrollar y simular un modelo epidemiológico usando SIG y dinámica de sistemas.	NU	ArcGIS	NE
Corner et al. [10]	Analizar, diseñar y construir el módulo de dispersión de deshechos marinos basado en técnicas de SIG	NU	TerrSet (IDRISI)	NU
Vairavamoorthy et al. [11]	Desarrollar un software predictor de riesgos asociados a los sistemas de distribución de agua fundamentados en SIG.	NU	ArcGIS	NU
Radiarta et al. [12]	Construir un modelo de evaluación multicriterio basado en técnicas SIG.	NU	ArcGIS	NU
Yu et al. [8]	Desarrollo de un sistema de recomendación a partir de la construcción de conocimiento colectivo.	H NE	UN	NU
Xu y Volker [13]	Desarrollar un GISSD en 3D para la evaluación de la sostenibilidad del desarrollo urbano residencial.	NU	ArcGIS	Vensim
Suarez et al. [14]	Diseñar y construir modelos y herramientas de localización óptima para franquicias comerciales.	NU	ArcGIS	NU

NU=No Utilizado **NE= No Especificado** **H=Híbrido** **BC=Basado en Conocimiento**

Artículo	**Objetivo**	**Sistema de recomendación**	**SIG**	**Dinámica de Sistemas**
Batet et al. [7]	Desarrollo un sistema de recomendación de películas híbrido para dispositivos móviles basado en agentes.	H NE	NU	NU
Noguera et al. [2]	Desarrollar e integrar un motor de recomendación móvil híbrido sensible a la ubicación con una arquitectura 3D-SIG.	H NE	NE	NU
Roig et al.[15]	Desarrollar un método de selección de puntos de ventas fundamentado en SIG y el proceso de jerarquía analítica.	NU	ArcGIS	NU
Casillas et al [16]	Aplicar y validar la técnica de modelado dinámico en la estimación de intensidad y hora en que ocurre la isla urbana de calor.	NU	TerrSet (IDRISI)	Stella
Li et al. [6]	Desarrollar un mecanismo de recomendación para compras grupales con cupones de descuentos a través de análisis de preferencia y ubicación geográfica.	H NE	NU	NU
Colombo et al [1]	Crear un sistema móvil de recomendación de funciones de películas; sensible a la ubicación, el tiempo y a los espectadores.	BC NE	NU	NU

Con base al análisis realizado en la presente se observa que existe la necesidad de desarrollar un sistema que permita ofrecer recomendaciones de puntos de venta integrando sistemas de recomendación con sistemas de información geográfica.

3. Arquitectura de integración

La arquitectura de integración presenta un enfoque basado en capas. Este tipo de diseño permite escalabilidad y mantenimiento debido a que sus tareas y responsabilidades se encuentran distribuidas. La Figura 1 presenta el esquema general de la arquitectura de integración propuesta. Cada capa tiene una función que se explica a continuación:

Capa de Presentación: La capa de presentación se encarga de actuar como medio de comunicación entre los resultados obtenidos por las demás capas para los usuarios.

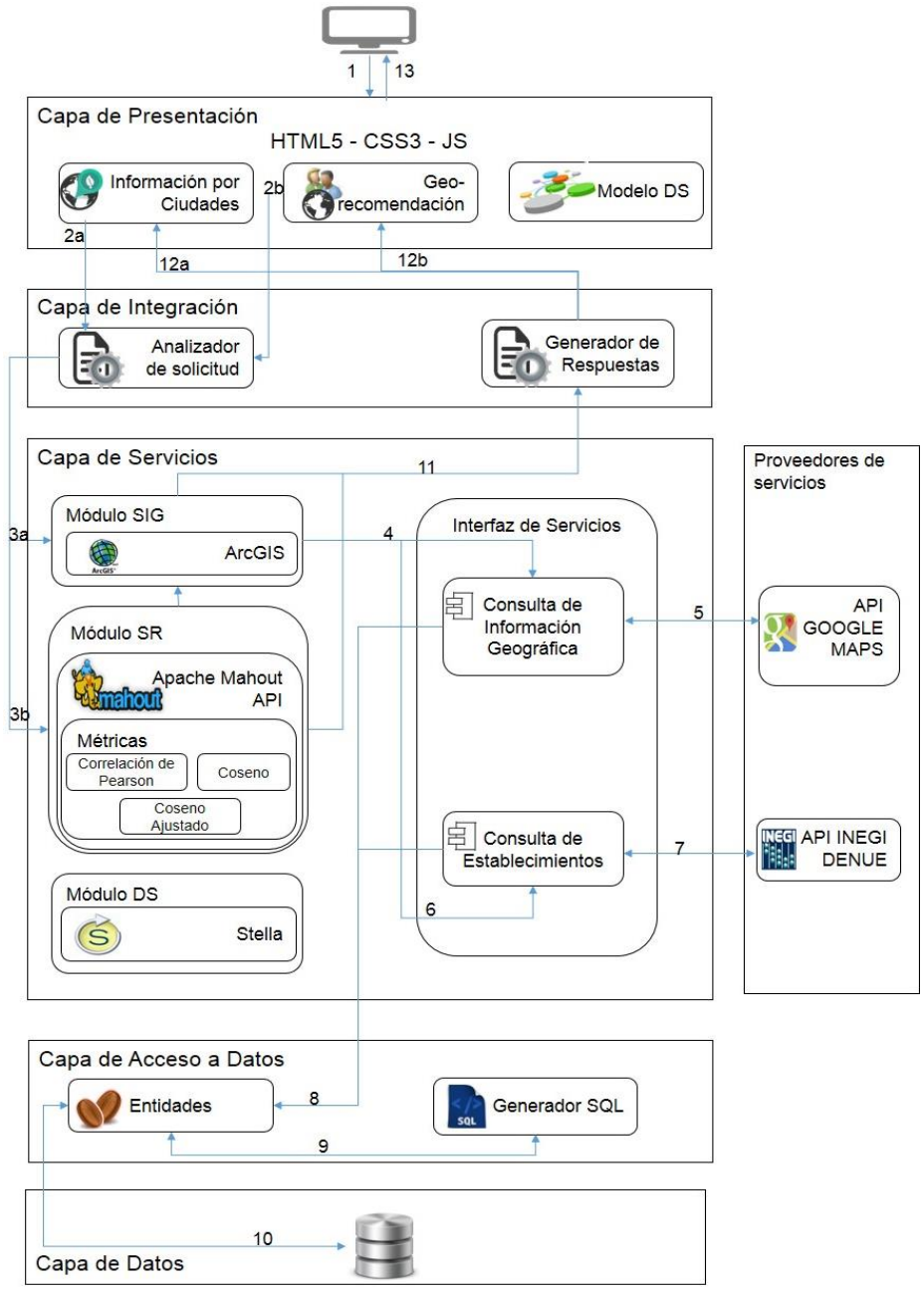


Fig. 1. Arquitectura del sistema de geo-recomendación.

Dentro de ella el usuario puede conocer la información de la ciudad, enviar la dirección del predio en forma de latitud y longitud para establecer un punto de venta;

además de mostrar la recomendación final en formato de mapa y ver el modelo de dinámica de sistemas. Actúa como la interfaz, en donde el usuario podrá enviar la dirección del predio y recibir la recomendación en formato de mapa.

Capa de Integración: Esta capa permite re direccionar las solicitudes a los servicios que fueron solicitados en la capa de presentación. Así como la construcción de las respuestas.

Capa de Servicios: En esta capa se encuentra gran parte de operaciones con las cuales funciona el sistema. En ella se hallan los módulos de SIG, de recomendación, de dinámica de sistemas e interfaz de servicios. Es importante mencionar que esta capa realiza el trabajo de ofrecer la recomendación así como también permite tener la información necesaria de las restricciones relacionadas con los puntos de venta.

Proveedores de Servicios: Dentro de esta capa se encuentran las entidades que presentan los servicios de identificación de establecimientos y Geolocalización proporcionados por las APIs INEGI DENUE y Google Maps. El API INEGI DENUE permite consultar datos de identificación, ubicación y actividad económica a nivel nacional, por entidad federativa y municipio [17]. Google Maps es una API que ofrece un servicio Web de aplicaciones de mapas que pertenece a Alphabet Inc. Google Maps permite obtener la ubicación geográfica a partir de la devolución de un radio preciso de localización [18].

Capa de Acceso de Datos: La capa de acceso de datos se encarga de buscar y guardar la información en la Base de Datos que le solicita la capa de servicios. Es posible la encapsulación de tareas a través de las distintas entidades y la ejecución de las operaciones de inserción, eliminación, consulta y actualización por medio del generador de instrucciones basadas en SQL.

Capa de Datos: Esta capa almacena información acerca de las poblaciones, así como sus asentamientos y establecimientos (cines, escuelas, orfanatos, asilos, hospitales, templos, guarderías, mercados, auditorios, estadios y teatros)

En esta arquitectura, cada módulo tiene una función bien definida la cual se describe a continuación:

Módulo SIG: Este módulo es responsable de construir el modelo geográfico con base en las características geográficas obtenidas a partir de la consulta de las características geográficas de la zona y de establecimientos que son solicitadas a la interfaz de servicios.

Módulo de Recomendación (RS): Es el responsable de ofrecer las sugerencias al correlacionar el perfil del usuario con respecto a otros perfiles. Así como también es responsable de evaluar que la ubicación de un punto de venta es la adecuada. Pide el histórico de las demás recomendaciones a la capa de datos y por medio de las métricas de correlación de Pearson, coseno y coseno ajustado es capaz de ofrecer la geo-recomendación.

Módulo de Dinámica de Sistemas (DS): Inicializa el modelo con los factores socioeconómicos para poder iniciar la simulación de posibles escenarios, generando información estadística.

Interfaz de servicios Web: Permite conectar con los servicios Web de geo-localización y de información de establecimientos; posibilita la creación de contenidos completos por medio de combinar datos provenientes de distintos servicios Web.

4. Caso de estudio: sistema de geo-recomendación

Para la validación de la arquitectura se plantearon los siguientes argumentos:

a) Una empresa desea establecer puntos de ventas en la ciudad de Orizaba, Ver. para comercializar su producto.
b) Es necesario que cada punto de venta cumpla con la regulación vigente perteneciente al Reglamento para las acciones de construcción, instalación, conservación y operación de estaciones de servicio en gasolinera y carburación.
c) La empresa desea automatizar el proceso de búsqueda de establecimientos para la colocación de puntos de venta.

Siendo estipulado lo anterior; ¿Cómo podría la empresa satisfacer las condiciones anteriores para establecer su punto de venta?

```
<Implies>
   <if>
      <And>
         <Atom>
            <op>
               <Rel>math:greaterThan</Rel>
            </op>
            <Var>distanciaResguardoRadialHospital</Var>
            <Var>distanciaResguardoRadialAsilo</Var>
            <Var>distanciaResguardoRadialOrfanato</Var>
            <Var>distanciaResguardoRadialGuarderia</Var>
            <Var>distanciaResguardoRadialEscuela</Var>
            <Data>300</Data>
         </Atom>
         <Atom>
            <op>
               <Rel>math:greaterThan</Rel>
            </op>
            <Var>distanciaResguardoRadialMercado</Var>
            <Var>distanciaResguardoRadialTemplo</Var>
            <Var>distanciaResguardoRadialAuditorio</Var>
            <Var>distanciaResguardoRadialEstadio</Var>
            <Var>distanciaResguardoRadialCine</Var>
            <Var>distanciaResguardoRadialTeatro</Var>
            <Data>150</Data>
         </Atom>
      </And>
   </if>
   <then>
      <Atom>
         <op>
            <Rel>predio</Rel>
         </op>
         <Data>adecuado</Data>
      </Atom>
   </then>
</Implies>
```

a)

```
<Implies>
   <if>
      <Or>
         <Atom>
            <op>
               <Rel>math:lessThan</Rel>
            </op>
            <Var>distanciaResguardoRadialHospital</Var>
            <Var>distanciaResguardoRadialAsilo</Var>
            <Var>distanciaResguardoRadialOrfanato</Var>
            <Var>distanciaResguardoRadialGuarderia</Var>
            <Var>distanciaResguardoRadialEscuela</Var>
            <Data>300</Data>
         </Atom>
         <Atom>
            <op>
               <Rel>math:lessThan</Rel>
            </op>
            <Var>distanciaResguardoRadialMercado</Var>
            <Var>distanciaResguardoRadialTemplo</Var>
            <Var>distanciaResguardoRadialAuditorio</Var>
            <Var>distanciaResguardoRadialCine</Var>
            <Var>distanciaResguardoRadialEstadio</Var>
            <Var>distanciaResguardoRadialTeatro</Var>
            <Data>150</Data>
         </Atom>
      </Or>
   </if>
   <then>
      <Atom>
         <op>
            <Rel>predio</Rel>
         </op>
         <Data>inadecuado</Data>
      </Atom>
   </then>
mplies>
```

b)

Fig. 2. a) Regla para predio adecuado, b) Regla para predio inadecuado.

Como primer paso, es necesario definir las reglas para la colocación de establecimientos las cuales se encuentran en el artículo 10 de la regulación

anteriormente mencionada; la cual indica que el predio debe ubicarse a una distancia mínima de resguardo de 300 metros radiales de centros de concentración masiva, tales como escuelas, hospitales, orfanatos, guarderías, asilos; así como a 150 metros radiales de mercados, cines, teatros, estadios, auditorios y templos [19]. El principal motivo de contar con reglas es el hecho de tener especificaciones que ayuden a identificar áreas óptimas. Para la representación de las reglas fue necesario utilizar RuleML 1.0. RuleML es un lenguaje para reglas en formato XML que provee una manera de expresar las reglas de negocio [20]; a través de la técnica de expresión de reglas se presentan las condiciones y acciones (antecedentes y consecuentes) que derivan de ella.

La Figura 2 presenta las reglas para la identificación de predios adecuados e inadecuados. Estas restricciones de ubicación indican que las estaciones de servicio deben respetar cierta distancia con respecto a museos, escuelas, hospitales, orfanatos, guarderías, teatros, cines, auditorios y templos. Como segundo paso, por cada asentamiento se buscó su información geográfica (latitud y longitud) por medio de la conexión al API de Google Maps. Cabe mencionar que para obtener la información respecto a los establecimientos mencionados anteriormente fue necesario construir una llamada al servicio proporcionado por la API DENUE del INEGI y guardar la información de los establecimientos, así como también la construcción de un mecanismo para la actualización de la información.

Por el momento se cuenta con un prototipo de la aplicación Web basada en el Framework JSF con PrimeFaces. La Figura 3, muestra la visualización de la información obtenida a partir de las consultas a los servicios Web de las APIs previamente mencionadas.

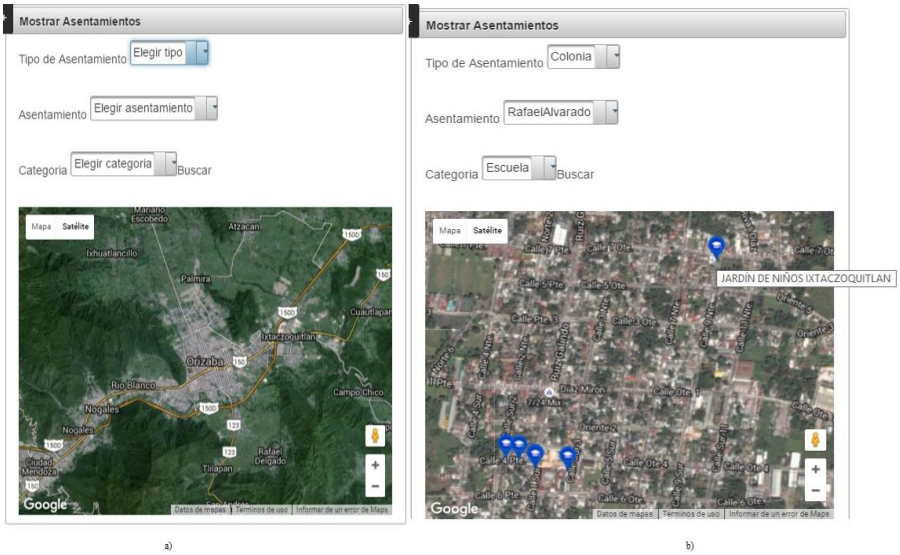


Fig. 3. Visualización de establecimientos.

Posteriormente, la Figura 4a muestra el formulario de búsqueda de puntos de venta, en él se introduce la dirección que se considera adecuada para establecer un punto de venta, en formato de longitud y latitud. En caso de no ser una ubicación

adecuada de acuerdo con las restricciones del reglamento, especificará que la localización no es la adecuada. La Figura 4b, presenta los resultados de sugerencia de posibles puntos de venta, en caso de que la ubicación si cumpla con las restricciones del reglamento; mostrando en forma circular de color verde los sitios más óptimos, en naranja los sitios con una probabilidad media y en rojo los sitios menos adecuados.

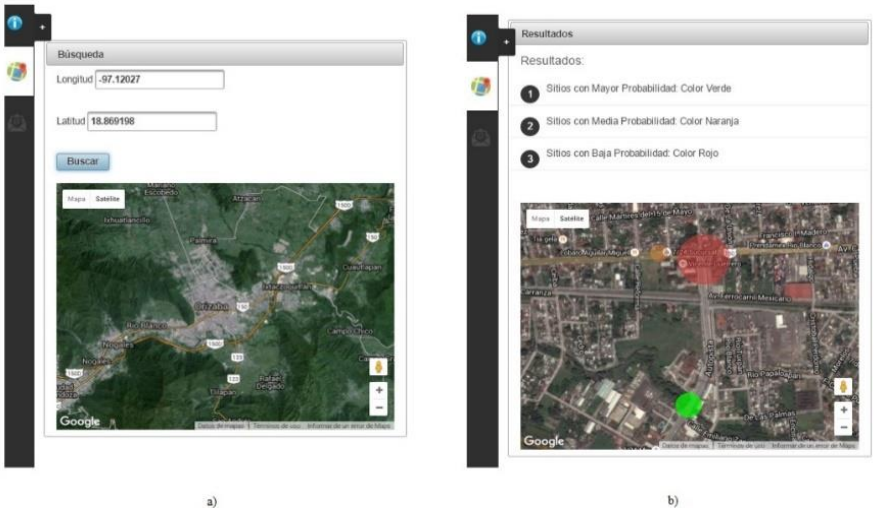


Fig. 4. a) Formulario de Búsqueda, b) Resultados.

Finalmente, la Figura 5 presenta el detalle de la forma circular seleccionada, donde un porcentaje nos indica que tan recomendable es la zona para establecer puntos de venta.

Fig. 5. Detalle de la zona.

Así como también la identificación de establecimientos cercanos (hospitales, asilos, guarderías, orfanatos, escuelas, mercados, templos, auditorios, cines, estadios y

teatros). El prototipo permite visualizar el detalle de cada una de las tres zonas (Alta, Media y Baja) encontradas mostrando los porcentajes y la puntualización de las reglas. Para este ejemplo se muestra el detalle de una zona con mayor posibilidad de ser adecuada para la colocación de un punto de venta; así como también la especificación de cumplir con las reglas de distancia de resguardo al no encontrase con ningún establecimiento cercano.

En esta sección se aprecia un solo escenario, no obstante se estan probando otros escenarios como la búsqueda radial, búsqueda por calle y búsqueda por codigo postal. La búsqueda radial permite identificar a partir de un punto (longitud y latitud) y una distancia radial ubicaciones apropiadas para establecer un punto de venta; donde muchas veces las superficies de estas búsquedas implican más de un asentamiento derivando en evaluaciones de restricciones para cada uno de los puntos que integren dicha superficie y en algunas ocasiones encuentra más de un punto de venta optimo. La búsqueda por calle de un asentamiento posibilita hacer una evaluación de las restricciones a todos los puntos que integran el tamaño de una calle. Esta es una búsqueda muy similar al caso de estudio salvo que en esta el usuario selecciona el asentamiento y la calle para poder encontrar un punto de venta.Finalmente la búsqueda por código postal hace evaluaciones por todas las superficies de los asentamientos que pertenecen al código postal y al igual que la búsqueda radial en determinadas circustancias devuelve más de un punto de venta adecuado. De manera que en casos con más de un punto de venta optimo, la decisión depende del usuario.

5. Conclusiones y trabajo futuro

El uso independiente de modelos geográficos y dinámica de sistemas para el análisis del establecimiento de locales comerciales, son las tradicionales líneas de investigación que están sujetas a limitaciones; ya que las características socio-demográficas influyen en la estrategia de localización de las empresas. La integración de los sistemas de información geográfica y modelos dinámicos facilita en gran parte la creación de estrategias que ayudan en la toma de decisión de las empresas. Sin embargo, esos estudios no ofrecen las características necesarias para el futuro, por lo que la incorporación de los sistemas de recomendación, dinámica de sistemas y SIG permitirá ayudar a los comerciantes en la toma de decisiones con respecto a la ubicación, tener la seguridad de encontrar un mercado potencial que les ayude a mejorar sus ingresos. Además de que la propuesta de la arquitectura de integración plantea la fusión de tecnologías que aún no han sido publicados por otros estudios.

Como trabajo futuro se pretende completar el desarrollo de los módulos de dinámica de sistemas, de recomendación y SIG, del sistema de recomendación para la ubicación de establecimientos que permita identificar las áreas de la zona de localización óptima. Así como también incrementar el número de ciudades pertenecientes a la zona centro del Estado de Veracruz y el estudio en la regulación vigente para la inclusión de reglas.

Agradecimientos. Los autores agradecen el apoyado por el Consejo Nacional de Ciencia y Tecnología (CONACYT), Tecnológico Nacional de México (TecNM) y la Secretaria de Educación Pública (SEP) a través de PRODEP. Así como también al

Instituto Nacional de Estadística e Informática (INEGI) por contar con APIs que permiten consultar la información del Territorio Mexicano.

Referencias

1. Colombo-Mendoza, L.O., Valencia-García, R., Rodríguez-González, A., Alor-Hernández, G., Samper-Zapater, J.J.: RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes. Expert Systems with Applications, Vol. 42, No. 3, pp. 1202–1222 (2015)
2. Noguera, J.M., Barranco, M.J., Segura, R.J., Martínez, L.: A mobile 3D-GIS hybrid recommender system for tourism. Information Sciences, Vol. 215, pp.37–52 (2012)
3. García-Palomares, J.C., Gutiérrez, J., Latorre, M.: Optimizing the location of stations in bike-sharing programs: a GIS approach. Applied Geography, Vol. 35, No. 1, pp. 235–246 (2012)
4. Zolfani, S.H., Aghdaie, M.H., Derakhti, A., Zavadskas, E.K., Morshed Varzandeh, M.H.: Decision making on business issues with foresight perspective; an application of new hybrid MCDM model in shopping mall locating. Expert systems with applications, Vol. 40, No. 17, pp. 7111–712 (2013)
5. Piastria, F.: Avoiding cannibalisation and/or conipetitor reaction en planar single facility location. Journal of the operations research, Vol. 48, No. 2, pp. 148–157 (2005)
6. Li, Y.-M., Chou, Ch.-L., Lin, L.-F.: A social recommender mechanism for location-based group commerce. Information Sciences, Vol. 274, pp. 125–142 (2014)
7. Batet, M., Moreno, A., Sánchez, D., Isern, D., Valls, A.: Turist@: Agent-based personalised recommendation of tourist activities. Expert Systems with Applications, Vol. 39, No. 8, pp. 7319–7329 (2012)
8. Yu, Y-H. Kim, J-H., Shin, K., Jo, G.S.: Recommendation system using location-based ontology on wireless internet: An example of collective intelligence by using 'mashup'applications. Expert systems with applications, Vol. 36, No. 9, pp. 11675–11681 (2009)
9. Castro Castro, C.A., Londoño Ciro, L.A., Valdés Quintero, J.C.: Modelación y simulación computacional usando sistemas de información geográfica con dinámica de sistemas aplicados a fenómenos epidemiológicos. Revista facultad de ingenieria Universidad de Antioquia, Vol. 34, pp. 86–100 (2005)
10. Corner, R.A., Brooker, A.J., Telfer, T.C., Ross, L.G.: A fully integrated GIS-based model of particulate waste distribution from marine fish-cage sites. Aquaculture, Vol. 258, No. 1, pp. 299–311 (2006)
11. Vairavamoorthy, K., Yana, J., Galgalea, H.M., Gorantiwara, S.D.: IRA-WDS: A GIS-based risk analysis tool for water distribution systems. Environmental Modelling & Software, Vol. 22, No. 7, pp. 951–965 (2007)
12. Radiarta, I.N., Saitoha, S-I., Miyazono, A.: GIS-based multi-criteria evaluation models for identifying suitable sites for Japanese scallop (Mizuhopecten yessoensis) aquaculture in Funka Bay, southwestern Hokkaido, Japan. Aquaculture, Vol. 284, No. 1, pp. 127–135 (2008)
13. Xu, Z., Coors, V.: Combining system dynamics model, GIS and 3D visualization in sustainability assessment of urban residential development. Building and Environment, Vol. 47, pp. 272–287 (2012)
14. Suárez-Vega, R., Santos-Peñate, D.R., Dorta-González, P.: Location models and GIS tools for retail site location. Applied Geography, Vol. 35, No.1, pp. 12–22 (2012)

15. Roig-Tierno, N., Baviera-Puig, A., Buitrago-Vera, J., Mas-Verdu, F.: The retail site location decision process using GIS and the analytical hierarchy process. Applied Geography, Vol. 40, pp. 191–198 (2013)
16. Casillas-Higuera, Á., García-Cueto, R., Leyva-Camacho, O., Gonzalez-Navarro, F.F.: Detección de la Isla Urbana de Calor mediante Modelado Dinámico en Mexicali, BC, México. Información tecnológica, Vol. 25, No.1, pp. 139–150 (2014)
17. http://www.inegi.org.mx/desarrolladores/denue/apidenue.aspx
18. https://developers.google.com/maps/documentation/geolocation/intro
19. Boley, H., Paschke, A., Shafiq, O.: RuleML 1.0: the overarching specification of web rules. Lecture Notes in Computer Science, 6403(4), pp. 162–178 (2010)
20. H. Ayuntamiento de Córdoba, Ver. Reglamento Para Las Acciones De Construcción, Instalación, Conservación y Operación De Estaciones De Servicio En Gasolinera y Carburación para el Municipio de Córdoba, Ver, pp. 3–4 (2014)

Síntesis óptima de un mecanismo de cinco barras de 2-GDL utilizando técnicas de inteligencia artificial

Maria Bárbara Calva Yáñez[1], Paola Andrea Niño Suárez[1], Jorge Alexander Aponte Rodríguez[3], Eric Santiago Valentín[2], Edgar Alfredo Portilla Flores[2], Gabriel Sepúlveda Cervantes[2]

[1]Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica Azcapotzalco (SEPI-ESIME-AZC), Cd. de México, México

[2] Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo, Cd. de México, México

[3]Universidad Militar Nueva Granada, Bogotá, Colombia

b_calva@hotmail.com, jorge.aponte@unimilitar.edu.co, e.santiag.valentin@gmail.com, {aportilla, pninos, gsepulvedac}@ipn.mx

Resumen. En este trabajo se presenta la síntesis dimensional óptima de un mecanismo de cinco barras para minimizar la fuerza en el eslabón de salida y asegurar el seguimiento de trayectorias en un espacio de trabajo predeterminado. Dicha síntesis se lleva a cabo al proponer un problema de optimización numérica con restricciones considerando aspectos estructurales y de fuerzas del mecanismo en estudio. El algoritmo denominado Evolución Diferencial (ED) es utilizado para resolver el problema de optimización resultante. Un conjunto de 30 ejecuciones independientes del algoritmo se lleva a cabo y el mejor de los resultados alcanzados es comparado con otras posibles soluciones. Se demuestra que el mecanismo óptimo tiene un mejor funcionamiento de acuerdo a la función objetivo seleccionado. Así mismo, se muestran resultados estadísticos paramétricos del rendimiento del algoritmo de ED, los cuales permiten sugerir un alto rendimiento del mismo para resolver problemas de diseño en ingeniería.

Palabras clave: Optimización, mecanismo plano, evolución diferencial.

Optimal Synthesis of a 2-DOF Five Bar Mechanism Using Artificial Intelligence Techniques

Abstract. This paper presents the optimal dimensional synthesis of a five -bar mechanism in order to minimize the force on the output link and ensure paths inside an established workspace. The synthesis is carried out by proposing a numerical optimization problem with constrains taking into account structural and force mechanism considerations. The Differential Evolution algorithm is

used to solve the resulting optimization problem. A set of 30 independent executions of the algorithm is carried out and the best of the results achieved compared with other possible solutions. It is shown that the optimal mechanism has a better performance according to the selected objective function. In addition, parametric statistical results of the DE algorithm performance are shown; such results suggest a high performance of the DE algorithm for solving engineering design problems.

Keywords: optimization, planar mechanism, differential evolution.

1. Introducción

La necesidad de incrementar el rendimiento de procesos industriales con la reducción de consumos de energía y/o recursos ha motivado a los expertos en computación a desarrollar algoritmos para resolver problemas duros de ingeniería, en razón a la alta complejidad que en muchos casos estos involucran. Es así como el planteamiento de problemas de optimización se ha convertido en una alternativa para solucionar estos problemas, permitiendo simplicidad de implementación, flexibilidad de aplicación y capacidad para encontrar buenas respuestas [1].

Los mecanismos planos han sido utilizados en forma extensiva para la solución de diversos problemas en la industria, como es la transmisión de movimiento. La necesidad de ajustar las trayectorias descritas por estos ha derivado en el uso de diferentes metodologías para la síntesis cinemática como son generación de función, trayectoria y movimientos [2]. La primera correlaciona la entrada con la salida del sistema, el segundo busca el control de un punto en el plano de modo que siga alguna trayectoria prescrita, en el tercero se realiza el control de una línea en el plano cuando esta asume algún conjunto de posiciones prescritas. Existen diferentes trabajos relacionados con la síntesis de mecanismos en los cuales a partir de la cinemática del sistema se plantean problemas de optimización numérica para obtener un conjunto de parámetros que describan el sistema, logrando que este se ajuste a un comportamiento requerido en aspectos como el seguimiento de trayectorias o minimización de fuerzas.

En [1] se realiza la síntesis dimensional óptima de un mecanismo de cuatro barras para garantizar la mayor transferencia de fuerza al eslabón de salida. En [3], el diseño de un efector final tipo pinza es realizado a partir de proponer un problema de optimización numérico con el objetivo de mantener la fuerza constante en todo el espacio de trabajo, dicho problema es resuelto utilizando una algoritmo de inteligencia colectiva. En [4] se realiza el diseño óptimo de un intercambiador de calor utilizando firefly como algoritmo. Los autores de [5] realizan la síntesis óptima para el seguimiento de trayectoria en mecanismos de cuatro barras utilizando programación cuadrática secuencial (SQP). Para el diseño de un efector final en el cual se mantiene la fuerza constante en el espacio de trabajo, [6] utiliza el algoritmo de Evolución Diferencial.

En diversas aplicaciones de ingeniería se requiere accionar mecanismos con el menor consumo de energía posible y con alta precisión en el posicionamiento, como son los sistemas automáticos de soldadura, corte o ensamble. En este trabajo se presenta la síntesis dimensional de un mecanismo de cinco barras en el cual además de lograr minimizar la fuerza en el eslabón de salida considerando aspectos dinámicos del

sistema, se ajusta el seguimiento de una trayectoria en un espacio de trabajo predeterminado, permitiendo ejecutar tareas como las ya mencionadas, aportando un enfoque en el cual se garantiza la menor fuerza en el actuador de salida para cualquier trayectoria contenida en el espacio de trabajo.

La organización del artículo es la siguiente: la Sección 2 describe el problema de síntesis de mecanismos, con una breve explicación de la cinemática, así como el análisis de fuerzas. En la Sección 3 se presentan las estrategias de optimización, analizando a la función objetivo y a las restricciones de diseño, además se presenta el planteamiento del problema de optimización. Posteriormente, en la Sección la Sección 4 se muestra el algoritmo empleado y algunos aspectos de la implementación computacional. Finalmente, en la Sección 5 se presentan y discuten los resultados obtenidos, y en la Sección 6 se exponen las conclusiones correspondientes.

2. Cinemática del mecanismo

El seguimiento de trayectorias en el plano ha sido solucionado con el uso de diferentes mecanismos, a continuación se presenta la síntesis de un mecanismo de cinco barras que minimiza la fuerza horizontal generada sobre el eslabón 5 y además facilita el ajuste paramétrico de trayectorias en un espacio de trabajo predeterminado.

2.1 Síntesis cinemática

El mecanismo mostrado en la Fig. 1, está compuesto por la barra 1 fija, mientras que la barra 2 realiza un movimiento de rotación pura. La barra 3 sirve de acoplador entre las deslizaderas 4 con movimiento de rotación y traslación (con la barra 2), y 5 la cual se traslada sobre la barra fija, siendo este el elemento de salida de la cadena cinemática. El punto de unión entre las barras 3 y 4 puede describir trayectorias en el plano.

Utilizando el criterio de Grubler – Kutzbach se determina la movilidad (m) del mecanismo (ecuación 1).

$$m = 3(n-1) - 2J_1 - J_2, \tag{1}$$

dónde:
n=número de eslabones
$J_1 =$ pares de un grado de libertad
$J_2 =$ pares de dos grado de libertad
Para este mecanismo:

$$m = 3(5-1) - 2(5) - 0 = 2. \tag{2}$$

Los resultados de la ecuación (2) indican que este mecanismo requiere dos movimientos independientes para que pueda realizar una trayectoria previamente establecida. Para realizar el análisis cinemático del mecanismo, este puede ser representado con el diagrama esquemático mostrado en la Fig. 2.

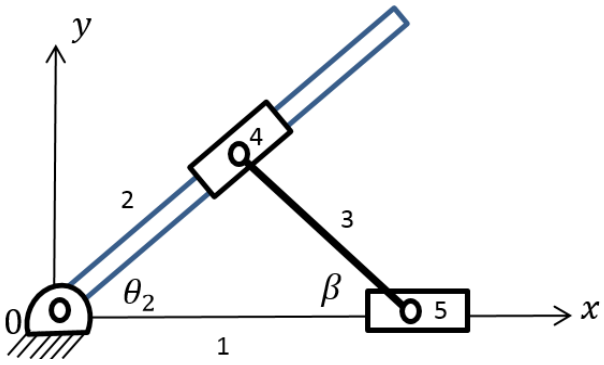


Fig. 1. Esquema general del mecanismo.

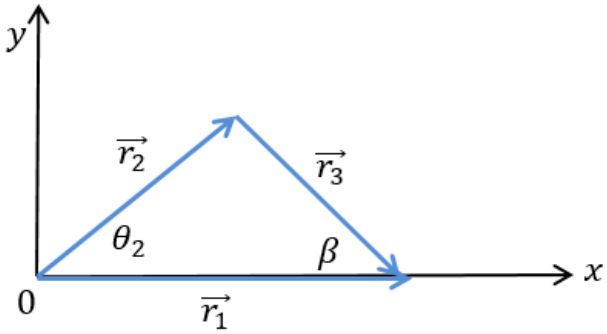


Fig. 2. Esquemático del mecanismo.

Se tendrá en cuenta que los vectores $\overrightarrow{r_1}$ *y* $\overrightarrow{r_2}$ cambian en magnitud, $\overrightarrow{r_3}$ permanece constante en magnitud y los ángulos θ_2 y β cambian, mientras que el eslabón 1 no cambia en dirección ($\theta_1 = 0$ constante). Para los análisis a continuación se hará $180 - \beta = \theta_3$

La ecuación de cierre del circuito es:

$$\overrightarrow{r_1} = \overrightarrow{r_2} + \overrightarrow{r_3}. \tag{3}$$

Puesta en forma polar:

$$r_1 e^{j\theta_1} = r_2 e^{j\theta_2} + r_3 e^{j\theta_3}. \tag{4}$$

Utilizando Euler:

$$r_1(cos\theta_1 + jsen\theta_1) = r_2(cos\theta_2 + jsen\theta_2) + r_3(cos\theta_3 + jsen\theta_3). \tag{5}$$

Separando la parte real e imaginaria de (5) y aplicando las condiciones de funcionamiento del mecanismo, se obtiene:

$$r_1 = r_2(cos\theta_2) + r_3(cos\theta_3), \tag{6}$$

$$\theta_3 = -arcsen((r_2 sen\theta_2)/r_3). \tag{7}$$

2.2 Análisis de fuerzas

Para el análisis de fuerzas se consideró el sistema en equilibrio, con una fuerza P, aplicada al eslabón 4. En la Fig. 3 se muestra el diagrama de cuerpo libre para este elemento.

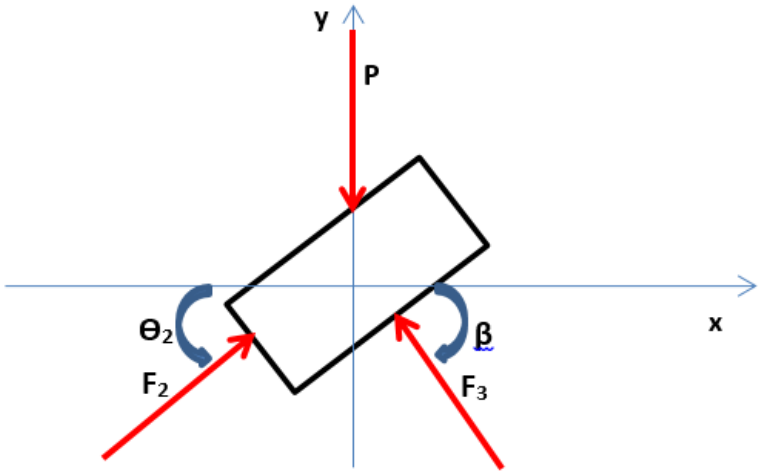


Fig. 3. Diagrama de cuerpo libre del eslabón 4.

En la Figura 3:
F_3= fuerza ejercida por el eslabón 3 sobre 4,
F_2= fuerza ejercida por el eslabón 2 sobre 4,
P = fuerza.

Con base en las leyes de Newton se plantean las ecuaciones de equilibrio:

$$\sum F_X = 0 \quad F_2 cos\theta_2 - F_3 cos\beta = 0, \tag{8}$$

$$\sum F_Y = 0 \quad F_2 sen\theta_2 + F_3 sen\beta - P = 0. \tag{9}$$

Resolviendo se llega a:

$$F_3 = \frac{P}{sen\beta + tan\theta_2 cos\beta}, \tag{10}$$

siendo

$$\beta = \frac{r_2}{r_3} sin\theta_2. \tag{11}$$

Para reducir los esfuerzos sobre el elemento 3 y minimizar la fuerza que se requiere al utilizar un accionamiento horizontal sobre el eslabón 5, se procede a elaborar el diagrama de cuerpo libre de este, ilustrado en la Fig. 4.

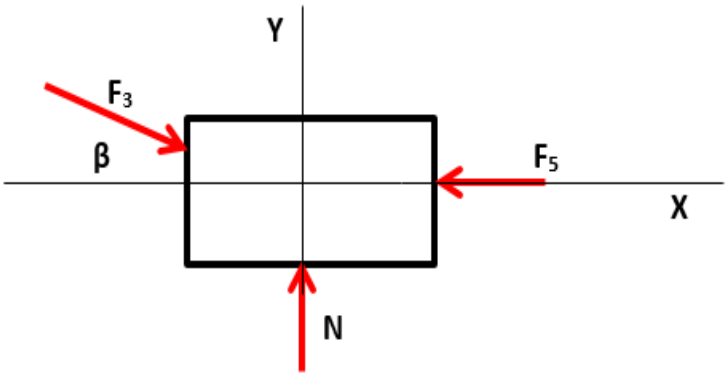


Fig. 4. Diagrama de cuerpo libre del eslabón 5.

Aplicando las condiciones de equilibrio se llega a:

$$\sum F_X = 0 \quad -F_5 + F_3 cos\beta = 0 \Rightarrow F_5 = F_3 cos\beta, \tag{12}$$

$$\sum F_Y = 0 \quad N - F_3 sen\beta = 0 \Rightarrow N = F_3 sen\beta. \tag{13}$$

Sustituyendo (10) en (12) y despejando se encuentra que la fuerza sobre el eslabón 5 está dada por

$$F_5 = \frac{P}{tan\beta + tan\theta_2} \tag{14}$$

3. Planteamiento del problema de optimización

Una vez realizados el análisis cinemático y de fuerzas, el diseño del mecanismo se plantea como un problema de optimización numérico, en el cual se deben especificar las relaciones matemáticas que permitan evaluar el desempeño del mismo.

3.1 Función objetivo

Como ya se planteó previamente, en este trabajo se desea determinar el tamaño de los eslabones que permitan seguir una trayectoria en un espacio de trabajo determinado por los puntos Ω y que además minimice la fuerza transmitida al eslabón de salida, siendo $\Omega = \{(10,10), (80,10), (80,50), (10,50)\}$, como se ilustra en la Fig. 5.

Considerando las ecuaciones obtenidas en el análisis de fuerzas, se plantea el problema de optimización numérica mono objetivo descrito por (14), para obtener la solución al problema de diseño para minimizar la fuerza sobre el eslabón 5:

$$\mathrm{f}(\beta_i, \theta_2^i) = \sum_{i=1}^{n} \frac{P}{sen\beta_i + tan\theta_2^i cos\beta_i}. \tag{15}$$

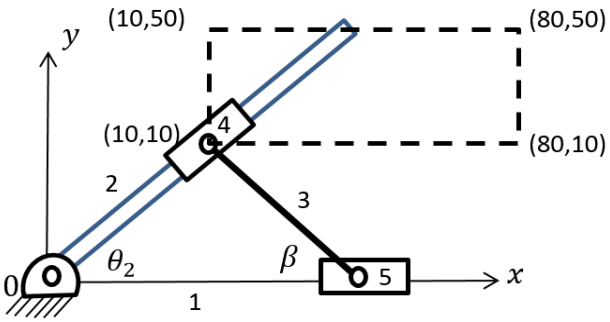


Fig. 5. Trayectoria a realizar por el punto de conexión entre los eslabones 4 y 3.

3.2 Restricciones de diseño

El cumplimiento de condiciones de funcionamiento está determinada por la dimensionalidad del mecanismo, el cual se establece como un conjunto de restricciones en el problema de optimización.

Para garantizar que el eslabón 4 pase por los puntos de precisión y que además $\beta \neq 90^0$ (evitando los problemas de montaje y funcionamiento que se generan cuando $\beta = 90^0$), se establece que $r_3 > r_2^i sen\theta_2^i$. Por otro lado al establecer que $r_3 sen\beta_i = y_i$, con $i = 1 \ldots 4$, se garantiza el paso del eslabón 3 por los puntos de precisión Ω

3.3 Vector de variables de diseño

Sea el vector de variables de diseño para el mecanismo de cinco barras establecido como:

$$\vec{x} = [r_3, \beta_1, \beta_2, \beta_3, \beta_4]^T = [x_1, x_2, x_3, x_4, x_5]^T, \tag{16}$$

donde r_3 es la longitud de la barra acopladora y $\beta_1, \beta_2, \beta_3, \beta_4$ son los ángulos que forma el eslabón 5 con respecto a la barra fija.

3.4 Problema de optimización

Teniendo en cuenta lo planteado en la sección 3.2, el problema de optimización numérica queda descrito por las ecuaciones 17 a 29.

$$\min_{x \epsilon \mathbb{R}^5} \sum_{i=1}^{n} \frac{P}{senx_2 + tan\theta_2^i cosx_2} \quad \text{n} \epsilon \mathbb{R}^5, \tag{17}$$

sujeto a:

$$g_1(\vec{x}) = r_2^1 sen\theta_2^1 - x_1 \leq 0 \tag{18}$$

$$g_2(\vec{x}) = r_2^2 sen\theta_2^2 - x_1 \leq 0 \tag{19}$$

$$g_3(\vec{x}) = r_2^3 sen\theta_2^3 - x_1 \leq 0 \tag{20}$$

$$g_4(\vec{x}) = r_2^4 sen\theta_2^4 - x_1 \leq 0 \quad (21)$$

$$h_1(\vec{x}) = x_1 senx_2 - Y_y^1 = 0 \quad (22)$$

$$h_2(\vec{x}) = x_1 senx_3 - Y_y^2 = 0 \quad (23)$$

$$h_3(\vec{x}) = x_1 senx_4 - Y_y^3 = 0 \quad (24)$$

$$h_4(\vec{x}) = x_1 senx_5 - Y_y^4 = 0. \quad (25)$$

Con cotas de:

$$0 < x_1 < 110, \quad (26)$$

$$0 < x_i < \frac{\pi}{2} \quad i = 2, \ldots, 5, \quad (27)$$

donde:

$$r_2^i = \sqrt{(x_i)^2 + (y_i)^2}, \quad (28)$$

$$\theta_2^i = tan^{-1}\left(\frac{y_i}{x_i}\right), \quad (29)$$

siendo P = 80 kg, y y_i la ordenada de los puntos de precisión Ω.

4. Algoritmo de optimización

El algoritmo de Evolución Diferencial (ED) es una de las heurísticas más utilizadas dentro de la computación evolutiva, ya que resuelve de manera eficiente los problemas duros de ingeniería [7]. ED parte de una población inicial de soluciones aleatorias y en cada generación se producen individuos de prueba aplicando operadores de cruza y mutación. La aptitud de cada nuevo individuo se evalúa compitiendo con un individuo de la población actual y así determina cuál de ellos se conservará para la siguiente generación. Las ventajas de que presenta ED, es el número reducido de parámetros de control tales como el tamaño de población (NP), el factor de cruza (CR) y el factor de mutación (F). Las características principales características de esta herramienta son:

- Representación de individuos,
- Selección de padres,
- Recombinación o cruza,
- Mutación y selección de sobrevivientes y variantes.

El diagrama de bloques correspondiente al algoritmo evolución diferencial se muestra en la figura 6.

Para el manejo de las restricciones se utilizan las reglas de DEB [8].

1. Entre dos individuos factibles, se escoge al que tenga la mejor función objetivo,
2. Entre un individuo factible y otro no factible, se escoge al factible,
3. Entre dos individuos no factibles, se escoge al que tenga un valor menor en la suma de violación a las restricciones.

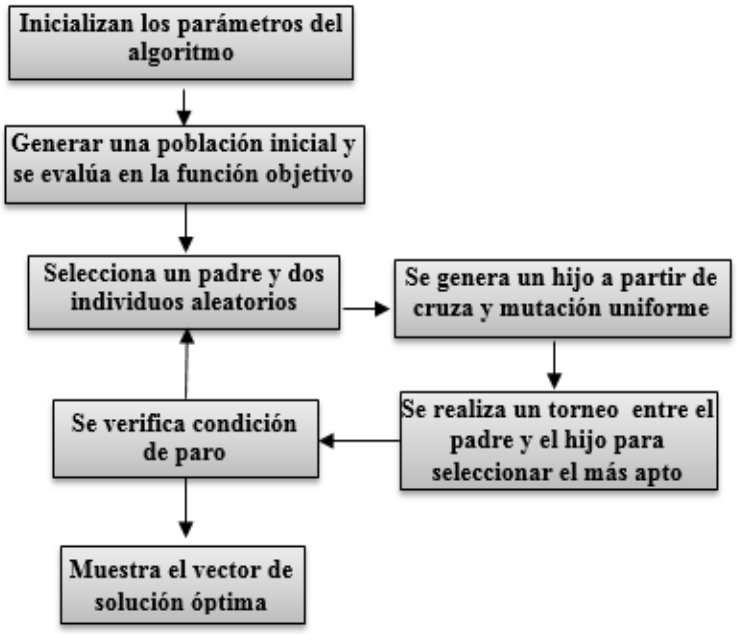


Fig. 6. Diagrama de bloques del algoritmo ED.

4.1 Implementación computacional

La implementación del algoritmo se realizó en Matlab R2013b®, ejecutado en una computadora con 8GB RAM, procesador INTEL CORE i7 a 2.2 GB y un sistema operativo Microsoft Windows 8. El programa se implementa en un módulo donde se calcula el valor de la función objetivo y las restricciones. Para el manejo de restricciones se realizan una sumatoria de violación de estas.

En la Tabla 1 se muestran las características del problema ***N*** es el número de variables del problema, **li** es el número de restricciones de desigualdad lineales, **ni** son las restricciones de desigualdad no lineales, **le** restricciones de igualdad lineales, **ne** las restricciones de igualdad no lineales, y $\boldsymbol{\rho}$ que es la complejidad del problema [9].

Tabla 1. Características del problema.

					ρ
5	4	4	0	0	0.000 %

El parámetro ρ mide la complejidad del problema, el cual es la relación entre la zona factible y el espacio de búsqueda, definido por Michalewicz y Shoenauer [10] como $\rho = |F|/|S|$, donde $|S|$ es el número total de soluciones generadas aleatoriamente y $|F|$ es el número de soluciones factibles obtenidas a partir de las soluciones generadas

aleatoriamente. Cuanto más disminuye ρ la complejidad para encontrar soluciones aumenta, representando un mayor reto para los algoritmos.

Para encontrar el valor del parámetro ρ de este caso de estudio se generan 1 millón de soluciones aleatorias, dentro de las cotas mencionadas, cabe resaltar que el problema tiene restricciones de igualdad y haciendo necesario el uso de un $\varepsilon=0.0001$.

5. Resultados

Se realizaron un conjunto de treinta simulaciones independientes con los siguientes parámetros: Un tamaño de población *NP* = 50, número máximo de generaciones *GMAX* = 4,999 con un numero de evaluaciones *EVAL* = 250,000 factor de cruza *CR* = [0.8, 1.0] por ejecución, y factor de escala *F* = [0.3, 0.9] por corrida. Los resultados obtenidos se muestran en la Tabla 2.

Tabla 2. Resultados de las 30 simulaciones.

Sim	r_3	β_1	β_2	β_3	β_4	*FO*
1	53.4658913	0.18814504	0.18814504	1.20874940	1.20874940	425.351273
2	53.4658914	0.18814504	0.18814504	1.20874940	1.20874940	425.351273
3	53.4658914	0.18814504	0.18814504	1.20874939	1.20874939	425.351273
4	53.4658915	0.18814504	0.18814504	1.20874939	1.20874939	425.351273
5	53.4658914	0.18814504	0.18814504	1.20874940	1.20874940	425.351273
6	53.4658913	0.18814504	0.18814504	1.20874940	1.20874940	425.351273
7	53.4658914	0.18814504	0.18814504	1.20874939	1.20874939	425.351273
8	53.4658914	0.18814504	0.18814504	1.20874940	1.20874940	425.351273
9	53.4658915	0.18814504	0.18814504	1.20874939	1.20874939	425.351273
10	53.4658914	0.18814504	0.18814504	1.20874940	1.20874940	425.351273
11	53.4658914	0.18814504	0.18814504	1.20874940	1.20874940	425.351273
12	53.4658914	0.18814504	0.18814504	1.20874939	1.20874939	425.351273
13	53.4658915	0.18814504	0.18814504	1.20874939	1.20874939	425.351273
14	53.4658913	0.18814504	0.18814504	1.20874940	1.20874940	425.351273
15	53.4658914	0.18814504	0.18814504	1.20874939	1.20874939	425.351273
16	53.4658914	0.18814504	0.18814504	1.20874939	1.20874939	425.351273
17	53.4658914	0.18814504	0.18814504	1.20874940	1.20874940	425.351273
18	53.4658914	0.18814504	0.18814504	1.20874940	1.20874940	425.351273
19	53.4658043	0.18814535	0.18814535	1.20875370	1.20875370	425.351273
20	53.2848938	0.18879087	0.18879142	1.21783176	1.21783536	425.370477
21	52.8198721	0.19047300	0.19047078	1.24256183	1.24256582	425.613713
22	52.8117989	0.19050330	0.19050192	1.24301109	1.24301236	425.619245
23	52.7549530	0.19071135	0.19070925	1.24619292	1.24619425	425.672008
24	52.7226199	0.19082724	0.19082669	1.24802558	1.24802304	425.705390
25	52.6434549	0.19111771	0.19111900	1.25255429	1.25254580	425.790906
26	52.5926037	0.19130702	0.19130610	1.25549575	1.25549232	425.852691

Sim	$\mathbf{r_3}$	$\boldsymbol{\beta_1}$	$\boldsymbol{\beta_2}$	$\boldsymbol{\beta_3}$	$\boldsymbol{\beta_4}$	***FO***
27	52.5121977	0.19160244	0.19160282	1.26022465	1.26022310	425.961349
28	52.1208640	0.19305699	0.19305955	1.28453759	1.28454012	426.699432
29	51.9761886	0.19360197	0.19360344	1.29415343	1.29415080	427.077478
30	51.6893145	0.19468957	0.19469123	1.31442892	1.31442442	428.044903

En la Tabla 2 se muestran los mejores vectores de solución obtenidos en cada simulación, ordenadas de manera ascendente, considerando el valor de función objetivo. Como se puede observar en las primeras 19 simulaciones se encuentra la misma solución, lo cual se debe la sintonización de parámetros del algoritmo que se realizó. En las 11 simulaciones restantes, se encuentran respuestas diferentes en razón a la alta complejidad del problema según el ρ calculado, lo cual repercute en el valor de la desviación estándar (0.6104) presentada en la Tabla 3.

Tabla 3. Estadísticas paramétricas.

Mejor	425.351272969054
Mediana	425.351272969054
Peor	428.044903376276
Promedio	425.636059394308
Desviación estándar	0.610428900392

En la Fig. 7a se muestra el comportamiento de los individuos que entran a la zona factible, donde se ilustran tres casos distintos, destacándose que antes de las 15,000 evaluaciones (300 generaciones) todos los ínvidos se encuentran en la zona factible, demostrando un buen desempeño del algoritmo. En la Fig. 7b, se realiza el análisis de convergencia para los tres casos representados anteriormente, encontrando que el caso promedio requiere menos evaluaciones para lograr la convergencia, mientras que el peor caso, no logra un mejor resultado aunque su población entró en menos evaluaciones a la zona factible

Las dimensiones obtenidas al solucionar el problema de optimización planteado, deben ser factibles y desde el punto de vista de ingeniería mecatrónica, de fácil manufactura y ensamble, lo cual se logra a partir de la mejor solución encontrada hasta el momento, con los resultados presentados en la Tabla 4 e ilustrados en la figura 8. La fuerza en cada una de las cuatro posiciones para el eslabón 3 también se especifica en dicha tabla, destacándose el mayor valor en la posición 2, a consecuencia de tener los eslabones con un ángulo pequeño con respecto a la horizontal.

Es posible proponer otras alternativas basadas en la experiencia del diseñador para el seguimiento de la trayectoria sin plantear un problema de optimización. De esta manera se puede considerar colocar al eslabón 3 a 90º del eslabón fijo mientras hace el recorrido en el espacio de trabajo de los puntos 3 a 4, lo cual hace que la sumatoria total de fuerzas transmitida a la deslizadera 5 sea de mayor que la obtenida al solucionar el problema de optimización planteado. Con un razonamiento similar se puede considerar una solución en la que la longitud de los eslabones 2 y 3 sean iguales en la posición 3, buscando una distribución uniforme de carga en dichos eslabones, de igual manera en estas condiciones la sumatoria total de fuerzas a lo largo de la trayectoria es mayor a la

mejor solución encontrada hasta el momento mediante el problema de optimización dichos resultados se pueden observar en la Tabla 5.

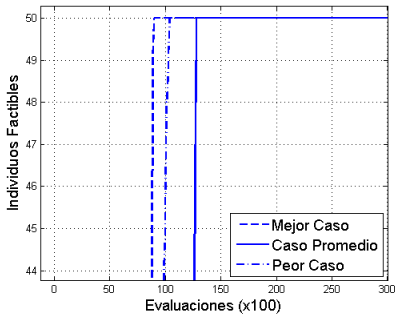


Fig 7a. Individuos factibles por evaluación.

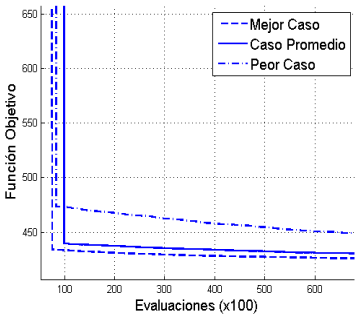


Fig 7b. Convergencia de la función objetivo.

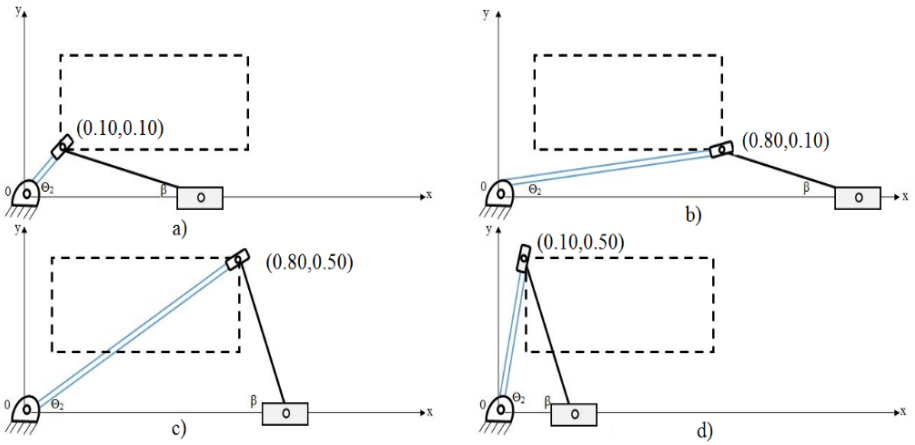


Fig 8. Posiciones alcanzadas por el mecanismo con las dimensiones encontradas.

Tabla 4. Dimensiones de la mejor solución encontrada hasta el momento.

Posición	r_1[m]	r_2[m]	r_3[m]	Θ_2 [Rad]	β [Rad]	F_3 [KgF]
(0.10,0.10)	0.62757554	0.14142135	0.53465891	0.78539816	0.18814504	68.411744
(0.80,0.10)	1.32522392	0.80622577	0.53465891	0.12435499	0.18814504	258.205195
(0.80,0.50)	0.99136380	0.94339811	0.53465891	0.55859932	1.20874940	69.1717183
(0.10,0.50)	0.46491057	0.50990195	0.53465891	1.37340077	1.20874940	29.562623

Tabla 5. Sumatoria de Fuerzas para diseños basados en la experiencia.

r_2[m]	r_3[m]	β [rad]	F_3 [KgF]
0.94339811	0.53465891	1.570793	513.5596216
0.50990195	0.50990195	0.197	432.5931608

6. Conclusiones

En este artículo se presentó la síntesis de un mecanismo de cinco barras para minimizar la fuerza en el eslabón de salida, asegurando el seguimiento de trayectorias dentro de un espacio de trabajo predeterminado con base en el algoritmo de evolución diferencial. De los resultados obtenidos se puede establecer que los algoritmos heurísticos son una herramienta viable para la solución de problemas duros de ingeniería, ya que permiten la reconfiguración del sistema a diseñar, modificando algunos parámetros del problema en el algoritmo.

La correcta interpretación del problema y la formulación de las restricciones, así como la sintonización de los parámetros del algoritmo requiere especial atención para la obtención de soluciones factibles y físicamente realizables. En los problemas de optimización que involucran restricciones de igualdad el manejo del parámetro ε influye en la obtención de buenas soluciones, teniendo en cuenta que este genera una margen de error aceptable, según la experiencia de quien resuelve el problema.

Con la solución encontrada se garantiza que la fuerza requerida por un accionamiento horizontal que mueva al eslabón 5, será mínima, logrando además reducir los esfuerzos generados sobre el eslabón acoplador 3. Al realizar un dibujó esquemático se comprueba que las dimensiones establecidas permiten alcanzar los puntos de precisión y que se pueden realizar trayectorias que estén contenidas en el espacio de trabajo que se genera con los cuatro puntos de precisión especificados, como se observa en la Fig. 8.

Agradecimientos. Los autores agradecen el apoyo recibido por el Instituto Politécnico Nacional a través de la Secretaría de Investigación y Posgrado vía los proyecto SIP 20161615, SIP 20161167 así como a los programas de EDI y COFAA. Los tres primeros autores agradecen al Consejo Nacional de Ciencia y Tecnología de México (CONACyT-México) por la beca otorgada para estudios de Posgrado en ESIME-AZCAPOTZALCO-IPN y CIDETEC-IPN, respectivamente. El segundo autor agradece a la Universidad Militar Nueva Granada por la comisión de estudios.

Referencias

1. Portilla Flores, E.A., Calva Yañez, M.B., Villareal Cervantes, M.G., Niño Suarez, P.A., Sepúlveda, C.G.: An Optimum Synthesis of a Planar Mechanism Using a Dynamic-based Approach. IEEE Latin America Transactions, Vol. 13, No. 5, pp. 1497–1503 (2015)
2. Norton, R.L.: Diseño de Maquinaria. Mc Graw Hill (2014)
3. Santiago Valentín, E.: Optimización de Sistemas Mecatrónicos utilizando herramientas de inteligencia artificial. Instito Politecnico Nacional IPN-CIDETEC, México (2016)
4. Dillip Kumar, M.: Application of firefly algorithm for design optimization of a shell and tube heat exchanger from economic point of view. International Journal of Thermal Sciences, Vol. 1, pp. 228–238 (2016)
5. Galeano Ureña, C.H., Duque Daza, C.A., Garzon Alvarado, D.A.: Aplicación de diseño óptimo dimensional a la síntesis de posición y velocidad en mecanismos de cuatro barras. Revista Facultad de Ingenieria Universidad de Antioquia, No. 47, pp. 129–144 (2008)

6. Valentin, E.S., Solano, P.A., Bautista, C.P., Rueda Melendez, J.M., Portilla Flores, E.A.: Diseño óptimo para transmisión de fuerza en un efector final. Research in Computing Science, No. 91, pp. 117–130 (2015)
7. Price, K.V.: An introduction to differential evolution. U.K. Maindenhead: Mc Graw - Hill (1999)
8. Deb, K.: An effcient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering, Vol. 186, pp. 311–338 (2000)
9. Shruti Goel, V.K.P.: Performance Evaluation of a New Modified Firefly. 3rd International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), Noida (2014)
10. Michalewicz, Z., Schoenauer, M., Koziel, S.: Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. Evolutionary Computation, Vol. 7, No. 1, pp. 19–44 (1999)
11. Yang, X.-S.: Firefly Algorithm, Levy Flights and Global Optimization. Research and Development in Intelligent System, Vol. 26, pp. 209–218 (2010)

Modelo de comportamiento de una turbina eólica

Uriel A. García, Pablo H. Ibargüengoytia, Alberto Reyes, Mónica Borunda

Instituto de Investigaciones Eléctricas,
Cuernavaca, Morelos, México

{uriel.garcia,pibar,areyes,monica.borunda}@iie.org.mx

Resumen. El comportamiento de un equipo puede verse como las variaciones en algunos de sus parámetros cuando existen cambios en el ambiente del equipo. El presente artículo muestra cómo el mecanismo de las redes Bayesianas pueden ser usadas para aprender un modelo probabilista del comportamiento. Con ese modelo, es posible la identificación de desviaciones al comportamiento normal. Es decir, realizar un diagnóstico en línea del equipo. Este artículo describe el modelo de comportamiento de una turbina eólica y los experimentos preliminares para identificar desviaciones al comportamiento normal. Se presentan experimentos para el aprendizaje del modelo usando datos históricos y experimentos de validación del modelo usando casos de estudio en que se presentó una falla en la turbina.

Palabras clave: Turbinas elicas, comportamiento, diagnstico, redes Bayesianas, SCADA.

Model of the Behaviour of Wind Turbine

Abstract. The behavior of an equipment can be seen as the variations of some parameters when changes in the envieronment are experienced. This paper describes how Bayesian networks can be used to learn a probabilistic model of the equipment's behavior. Using this model, it is possible to identify deviations to the normal behavior. This means that on–line diagnosis can be executed. This paper describes the behaviour model of a wind turbine and the preliminary experiments to identify deviations. The experiments to learn the model based on historical data are presented and the experiments to validate the model using a turbine failure data.

Keywords: Wind turbines, behavior, diagnosis, Bayesian networks, SCADA.

1. Introducción

El diccionario de la Real Academia Española define al comportamiento como sigue:

> Comportamiento o conducta es la manera de proceder que tienen las personas u organismos, en relación con su entorno o mundo de estímulos. El comportamiento puede ser consciente o inconsciente, voluntario o involuntario, público o privado, según las circunstancias que lo afecten.

En el caso de procesos o sistemas industriales, la definición incluiría la manera de proceder del sistema de acuerdo a su entorno y a los estímulos que reciba. Entonces, se puede describir el comportamiento de un proceso industrial como las variaciones que tienen sus variables principales dadas las variables de entrada al proceso y el entorno.

El trabajo de investigación reportado en este artículo pretende la modelación del comportamiento de una turbina eólica. Se requiere entonces modelar el proceder adecuado de la turbina de acuerdo al entorno suyo y a los estímulos que puede recibir. Entonces, el modelo deberá indicar los cambios que se registran en las partes de la turbina cuando se resienten cambios en el entorno. En otras palabras, algunas variables tendrán valores diferentes cuando otras variables cambien de valor.

Para el caso de la turbina eólica se decidió tomar a las variables que se registran en el sistema de control supervisorio y de adquisición de datos (SCADA por sus siglas en inglés) como las variables que caracterizarán el comportamiento de la turbina. Se propone el uso de redes Bayesianas (RB) para la generación de modelos de comportamiento de equipo industrial. Las RB codifican las relaciones probabilistas entre las variables del equipo. Se puede identificar las variables dependientes entre ellas, las variables independientes y las relaciones condicionales entre varias variables. Formalmente, las RB son grafos acíclicos dirigidos que codifican relaciones probabilistas entre variables. Los nodos de una RB representan a las variables mientras que los arcos dirigidos representan relaciones probabilistas. El nodo destino de un arco (el hijo) es dependiente del origen del arco (el padre).

La RB aprendida se considera exclusivamente como un modelo que codifica el comportamiento de la turbina. Dados los valores de unas variables, podremos calcular las probabilidades del valor de otras variables. Entonces, comparando los valores estimados con los valores reales observados en el SCADA, podremos identificar desviaciones a ese comportamiento. Esta idea sigue la teoría de la validación de sensores desarrollada anteriormente [3]. Aunque en aquel trabajo se buscó identificar cuando un sensor mandaba información incorrecta, se desarrolla en este proyecto la idea que, al detectar información inesperada, el sensor esté funcionando correctamente pero el equipo no está comportándose correctamente.

Este artículo se organiza como sigue. La siguiente sección introduce brevemente el formalismo de las redes Bayesianas. La sección 3 explica las principales características de una turbina eólica. Sus partes que la componen y algunas de las variables que se manejan en el sistema SCADA. En seguida, la sección 4

expone la construcción del modelo del comportamiento de la turbina y muestra la red Bayesiana resultante. En seguida, se explica la forma de identificar la desviación al comportamiento normal representado por el modelo Bayesiano en sección 5. La sección 6 describe los experimentos que se han realizado para la validación del modelo de comportamiento de la turbina, así como su capacidad para identificar fallas o desviaciones al comportamiento normal. Finalmente, la sección 7 concluye el artículo, discute los resultados y muestra la dirección que este trabajo de investigación llevará en el futuro.

2. Introducción a las redes Bayesianas

Las redes Bayesianas (RB) [5] fueron creadas para representar incertidumbre en sistemas inteligentes. Fueron desarrolladas con la conjunción de teoría de probabilidad con la teoría de grafos. Por definición, las RB son grafos acíclicos dirigidos que representan las relaciones probabilistas entre las variables del proceso. Los nodos representan las variables u objetos del dominio de aplicación, mientras que los arcos representan las relaciones probabilistas entre los nodos enlazados. El conocimiento es representado en RB de dos formas:

Cualitativamente con la estructura de la red. Se muestra cuáles nodos son dependientes y cuáles son independientes.

Cuantitativamente con los parámetros que forman las probabilidades a-priori de los nodos raíz y las matrices de probabilidad condicional de los nodos hijos dados los valores de los nodos padres.

La estructura se adquiere utilizando conocimiento de expertos que puedan indicar las relaciones probabilistas entre variables. Sin embargo es difícil tener ese conocimiento en aplicaciones reales. Para esto, existen una buena variedad de algoritmos de aprendizaje automático que proporcionan la estructura de RB en base a datos históricos del proceso [4].

Los parámetros también pueden ser proporcionados por expertos en el tema o aprendidos automáticamente con los algoritmos apropiados. Juntando las características del problema de la representación del conocimiento con las características de las RB, éstas resultan un magnífico enfoque para modelar ese conocimiento. La Fig. 2 muestra un ejemplo de modelo para la representación del comportamiento de una turbina eólica.

3. Turbinas eólicas

Las turbinas eólicas se nombran en base a la orientación del eje de pala de rotor en horizontales o verticales. El viento se aprovecha y se convierte en electricidad mediante las turbinas eólicas, la cantidad de electricidad que produce una turbina depende de su tamaño y la velocidad del viento. Las turbinas de eje horizontal se caracterizan por compartir diseño y partes básicas: palas, rotor, eje central, multiplicadora, generador, torre, góndola, freno. Estas partes trabajan

en conjunto para convertir la energía cinética del viento en energía mecánica que genera electricidad. La Fig. 1 muestra un diagrama de partes de la turbina marca Komai.

Las secciones de una turbina se describen a continuación:

Rotor: Se encuentra conectada al eje principal, elemento rotatorio para recibir la energía a partir del viento mediante sus tres palas conectadas al buje del rotor.

Soportes o rodamientos: Se cuenta con soportes para el sistema yaw, pitch, eje principal del rotor y el rotor que son monitoreados conforme a su temperatura.

Sistema de orientación: Los aerogeneradores cuentan con dos sistemas de orientación en relación a la dirección del viento. El primer sistema se refiere a la posición de la góndola (yaw) y el segundo sistema es la posición adecuada de las paletas (pitch) con la cual se controla la velocidad del rotor.

Sistema de enfriamiento: Elemento indispensable para evitar las altas temperaturas en el interior de la góndola y disminuir las condiciones extremas de sus componentes.

Sistema de frenado: Dispositivo de bloqueo que frena la rotación del rotor poniéndolo en un punto muerto. El evento se genera de manera manual o automático por ejemplo en caso de mantenimiento o exceso de viento.

Mástil meteorológico: Está equipada para medir velocidad y dirección del viento con un alto grado de precisión y transmite la información al controlador. Cuenta con anemómetros y veletas rodeado de tubos de metal que funcionan como pararrayos.

Generador: Se encuentra conectada a la caja multiplicadora la cual convierte la energía mecánica en energía eléctrica produciendo electricidad.

Multiplicadora de velocidad: Conecta el eje de baja velocidad al eje de alta velocidad e incrementa la velocidad rotacional del generador requerida para producir electricidad. Por ejemplo en la turbina Komai, la relación de transmisión es de un aumento rotacional nominal del rotor de 40.5 rpm a 1795 rpm de la rotación del generador.

Góndola: Se encuentra sobre la torre y contiene la caja multiplicadora, eje de baja y alta velocidad, generador, controlador, sistema de frenado etc. Es monitoreado por vibración y temperatura.

SCADA es una aplicación de software para el control y monitoreo con los datos medidos por PLC, el cual se encarga de la comunicación con los dispositivos instalados en el aerogenerador para la adquisición de datos. El software se encarga de almacenar, procesar, supervisar y controlar la información recibida a través de su IHM (interfaz hombre maquina) el cual permite modificar parámetros de funcionamiento de la turbina eólica. LA IHM del SCADA permite visualizar y adquirir una serie de variables que describen el comportamiento de los sensores instalados en la turbina apreciando el nombre de la variable, valores medidos en promedios de 5 minutos, desviaciones estándar, valores mínimos y máximos.

La Tabla 1 muestra un ejemplo de variables correspondientes a los diferentes equipos que forman la turbina. Se muetran tres variables de tres equipos.

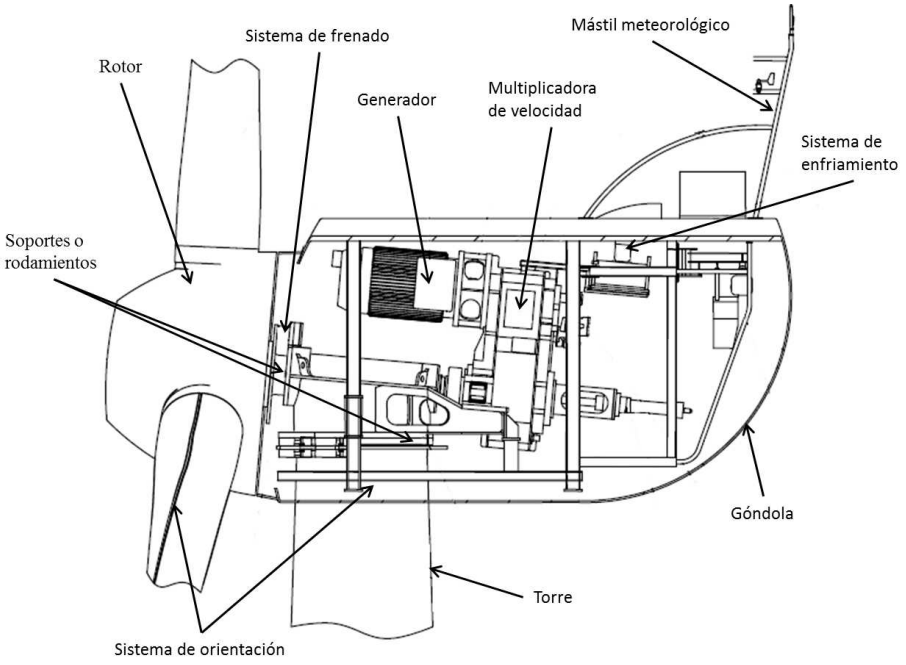


Fig. 1. Modelo de la turbina marca Komai.

4. Representación del comportamiento de turbinas eólicas

Para el aprendizaje del modelo probabilista del comportamiento de la turbina eólica, se utilizó uno de los algoritmos de aprendizaje de redes Bayesianas incluidas en el paquete *Hugin* [1]. Específicamente, utilizamos el algoritmo *Gready and search* [2]. Para ese aprendizaje, se siguió el procedimiento a continuación:

1. Identifica un conjunto de datos históricos en que un experto pueda sugerir comportamiento normal de la turbina. Utiliza el formato de datos separados por coma (csv). Mientras más grande este conjunto de datos mejor, ya que se presentarán diferentes condiciones de operación de la turbina.
2. Detecta los datos inconsistentes (*outliers*) y elimínalos del archivo de datos.
3. Discretiza el conjunto de datos utilizando partición uniforme. El número de intervalos depende del poder de cómputo disponible. A más intervalos, se aprenden modelos más caros computacionalmente.
4. Corre el algoritmo de aprendizaje de redes Bayesianas usando el paquete Hugin. Depura el modelo. Esto es, si al generar la red Bayesiana resulta en nodos aislados de la red, elimínalos del conjunto de datos y repite los pasos anteriores.

Fig. 2 muestra el modelo aprendido. Nótese por ejemplo algunas relaciones que se perciben lógicas cómo las relaciones entre los nodos $WindSpeed$ y

Tabla 1. Ejemplo de diferentes equipos que forman la turbina y sus variables

Sistema de orientación			Generador			Rotor		
Variable	Un.	Descripción	Variable	Un.	Descripción	Variable	Un.	Descripción
Pitch Angle	◦	Regula el ángulo de paso de las aspas del rotor	Bearing temp DE	◦	Temperatura interior de chumacera	Bearing Temp A DriveEnd	◦	Temperatura de chumacera en posición A
Pitch Pressure	bar	Control de presión del ángulo de paso de las aspas	Bearing temp NDE	◦	Temperatura exterior de chumacera	Bearing Temp B NoDriveEnd	◦	Temperatura de chumacera en posición B
Brake Pressure	bar	Presión de frenado	Generator Speed	rpm	Velocidad del generador	Rotor Speed	rpm	Velocidad del rotor

$InverterMotorCurrent$, $GeneratorSpeed$ y $Vibration$. Es decir, si existe una velocidad de viento alta, se espera que la velocidad del generador esté alta, que las vibraciones aumenten así como la corriente en el inversor. De igual manera, si hay poco viento, las demás variables se comportarán con valores bajos. La intensidad de estas relaciones está dada por las tablas de probabilidad condicional que conforman el modelo completo. Con este modelo, se puede hacer una estimación probabilista y comprobar si ambas cantidades están de acuerdo. En caso contrario, se detecta una desviación al comportamiento normal que habrá que profundizar. La siguiente sección describe ese proceso.

5. Detección de desviaciones al comportamiento normal

Las RB representan un enfoque apropiado para el modelado del comportamiento de las turbinas eólicas. Sin embargo, lo importante será la detección de desviaciones del comportamiento normal. Esto se realiza utilizando la teoría de validación de sensores desarrollada anteriormente [3]. En objetivo de esa teoría es la identificación de fallas en sensores en un proceso industrial. La idea es la identificación de valores atípicos de una variable de acuerdo a los valores de las variables más relacionadas. Sin embargo, en este artículo desarrollamos la idea de que un sensor puede marcar como erróneo pero no por falla del sensor sino por desviación del comportamiento del proceso.

Esta teoría se desarrolla en dos fases:

1. detección de una desviación al comportamiento normal y
2. aislamiento o identificación de la variable con falla.

La fase de detección se realiza en un ciclo donde cada variable es estimada en la RB y comparada con su valor real. Para esto, se asigna valores a

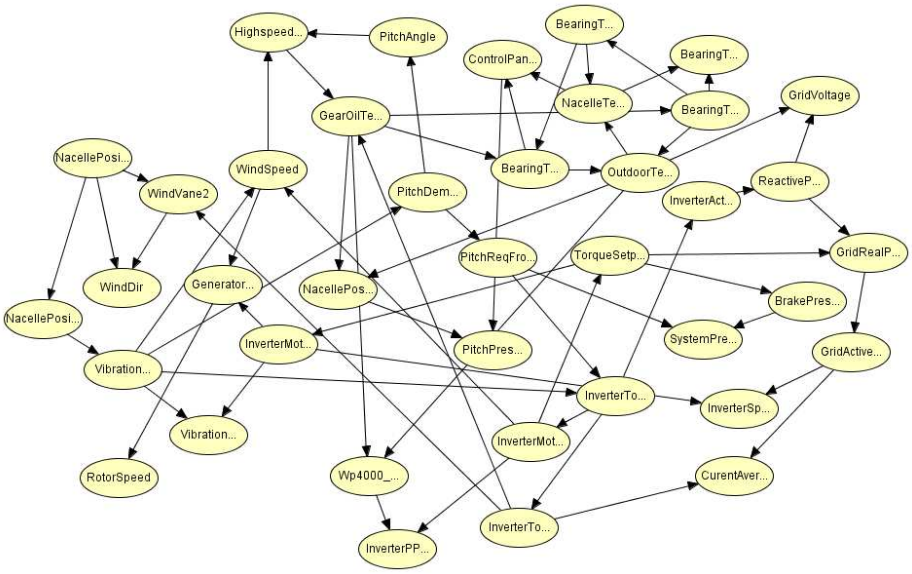


Fig. 2. Modelo de comportamiento aprendido.

las variables relacionadas a la variable en observación y se propaga las probabilidades del nodo. En caso de existir una diferencia importante, se declara una falla aparente o potencial. La fase de aislamiento recibe el conjunto de nodos con fallas aparentes generada en la fase de detección e identifica cuál es la falla real. Esta identificación se realiza utilizando la propiedad de la cobija de Markov de una RB [5]. La cobija de Markov de un nodo en una red Bayesiana está formada por el conjunto de los nodos padres, más los nodos hijos más los nodos esposos, es decir, los otros padres de los hijos. Por ejemplo, en la Fig.2, la cobija de Markov del nodo $WindSpeed$ son los nodos $\{Vibration, InverterMV, Generator, Highspeed, PitchAngle, InverterMC\}$. En suma, después del ciclo de detección, el nodo cuya cobija de Markov coincida con el conjunto de fallas aparentes, entonces esa variable es la que provocó la falla real. Sin embargo, en este proyecto de investigación se pretende identificar no solo las fallas de sensores individuales sino además, desviaciones del comportamiento normal de la turbina. La siguiente sección describe los primeros experimentos en esa detección de desviación del comportamiento.

6. Experimentos de validación

Se utilizaron 39 variables de la base de datos del sistema SCADA para la construcción del modelo de comportamiento de una turbina eólica incluyendo mediciones de temperatura, presión, vibración, posición, velocidades, potencia, corriente entre otras involucradas en los diferentes equipos de una turbina.

La Tabla 2 muestra las 39 variables utilizadas con sus nombres como vienen en el SCADA. Se menciona las unidades de las variables.

Tabla 2. Lista de variables usadas en el modelo de comportamiento

Nombre de variable	Unidad	Nombre de variable	Unidad
GridRealPower100ms	kW	NacellePosition	◦
GridActivePower	kW	PitchReqFromPitchControleller	◦
TorqueSetpoint	Nm	PitchAngle	◦
CurentAverage3phase_32bit	A	PitchDemand	◦
ReactivePower1	kVar	NacellePositionRelativeToWindDirection	◦
WindSpeed	m/s	NacellePositionRelativeToWindDirection30s	◦
Vibration1WP4084_1	G	WindDir	◦
Vibration2WP4084_1	G	WindVane1	◦
PitchPressure	bar	Wp4000_Temp	◦ C
SystemPressure	bar	BearingTemp_A	◦ C
BrakePressure	bar	BearingTemp_B	◦ C
RotorSpeed	rpm	GearOilTemperature	◦ C
GeneratorSpeed	rpm	HighspeedBearingTemp	◦ C
InverterMotorVoltage	rpm	Bearing temp DE	◦ C
InverterTorqueReference	%	BearingTempNDE	◦ C
InverterTorque	%	InverterPP1Temperature	◦ C
InverterActivePower	%	OutdoorTemperatureLog	◦ C
InverterMotorVoltage	V	ControlPanelTemperature	◦ C
InverterMotorCurrent	V	NacelleTemperature	◦ C
GridVoltage	V		

Para la generación del modelo de comportamiento se utilizó un conjunto de datos limpios, el modelo se construyó con un año de información adquirida a cada 5 minutos en condiciones normales de operación de la turbina eólica. Se consideró para el conjunto de prueba datos con condiciones de operación irregulares.

Usando las 39 variables y la información histórica de un año, se construyó el modelo de comportamiento mostrado en la Fig.2. Con esa red Bayesiana, se realizaron los siguientes experimentos:

Para el obtener los resultados presentados en la Fig. 3. Se siguió el procedimiento a continuación:

1. Identificar el conjunto de variables en la base de datos del SCADA. El conjunto de datos de pruebas utiliza el formato de datos separados por coma (csv).
2. Seleccionar datos con comportamiento inconsistente en la generación de potencia.
3. Correr el modelo de comportamiento con el conjunto de datos de prueba.
4. Visualizar resultados de fallas reales.

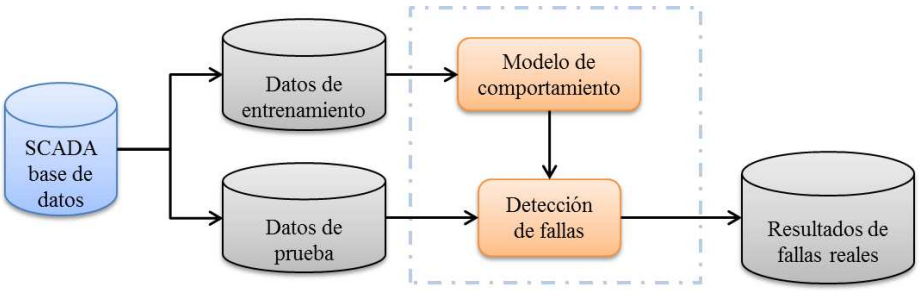


Fig. 3. Diagrama a bloques del procedimiento de validación.

La Fig. 4 muestra una sección de los datos seleccionados para la validación del modelo. Se muestran únicamente las variables de velocidad del viento (WindSpeed) y de la potencia generada (GridActivePower). Se puede apreciar un momento en que la potencia generada tuvo una leve caída y más adelante, se puede apreciar una caída total de la generación. Se puede suponer que en estos momentos sucedió algún evento que forzó al cambio de comportamiento de la turbina. La Fig. 5 muestra la detección de comportamiento anormal de algunas variables. En el eje vertical se puede apreciar la probabilidad de falla detectada en esas variables.

Se observa en los resultados las fallas evidentes de las siguientes variables: Bearing temp DE (temperatura interior de soportes del generador), Bearing-Temp_A (temperatura de soportes del eje principal), GridVoltage (Voltaje en la Red), NacellePosition (Posicion de la gondola), ControlPAnelTemperature (Temperatura de panel de control de la turbina eólica).

Sobreponiendo las dos figuras, se hace contraste a la generación de potencia de la turbina eólica donde se apreciar caídas de potencia e instancias de inestabilidad en la generación. Esto valida la capacidad del modelo para la detección de comportamiento anormal. Cuando se presenta algún evento inesperado, que puede ser falla en algún componente de la turbina, se genera un patrón de fallas en algunas variables del modelo. Ese patrón de variables en falla se clasificará para poder hacer un diagnóstico completo y en línea de la turbina eólica.

7. Conclusiones y trabajo futuro

Se muestra en este artículo los avances del proyecto que persigue el diagnóstico inteligente de turbinas eólicas. El enfoque que se utiliza es el de modelado del comportamiento de la turbina para poder identificar desviaciones a éste. Se describe en el artículo el enfoque para crear el modelo de comportamiento usando redes Bayesianas y se describen los experimentos iniciales que muestran que es posible la detección de comportamiento anormal. Se utilizó un conjunto de datos para la prueba donde se puede apreciar dos comportamientos sospechosos y cómo se muestran en el sistema. Sin embargo, el trabajo seguirá atendiendo las siguientes preguntas abiertas:

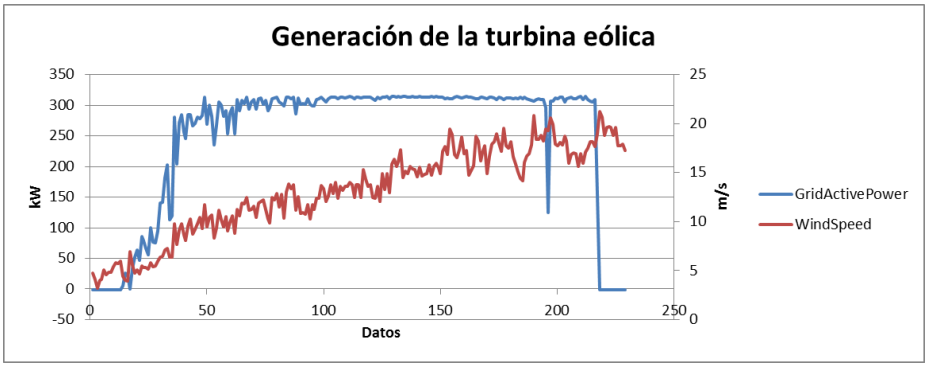


Fig. 4. Gráfica que muestra el comportamiento de la velocidad del viento y la generación de potencia.

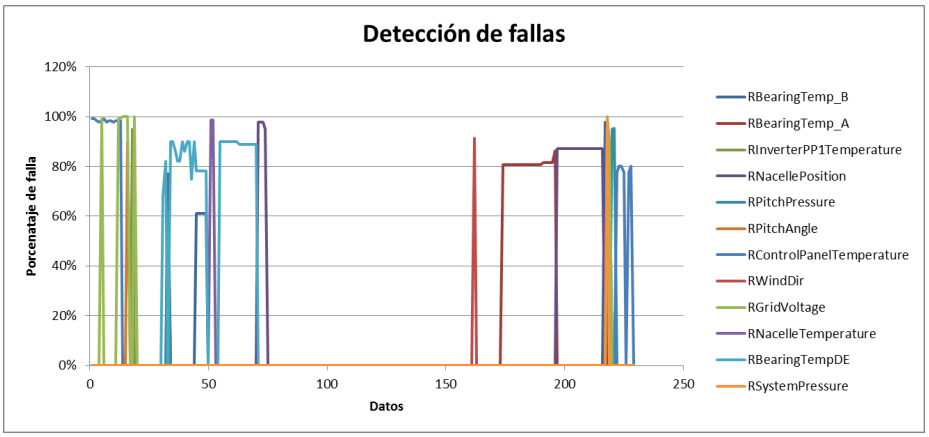


Fig. 5. Gráfica que muestra la identificación de patrones de falla de algunas variables.

- se podrá identificar modelos de comportamiento más compactos según las diferentes partes de la turbina? Esto generaría un conjunto de modelos que podrán estar interconectados formando redes Bayesianas multi-seccionadas [6].
- según la definición de comportamiento, se podrá identificar diferentes contextos del ambiente para analizar el comportamiento apropiadamente en cada contexto? No es lo mismo el comportamiento de la turbina con bajos vientos que con los vientos de mayor velocidad como los de la región de Oaxaca.
- se requerirá de la identificación de patrones de desviaciones detectadas para caracterizar fallas específicas de la turbina?
- será posible detectar esas fallas de manera insipiente?

El proyecto seguirá avanzando para responder las preguntas planteadas.

Agradecimientos. El trabajo reportado en este artículo es financiado por la Secretaría de Energía y el CEMIE-Eólico del Fondo Sectorial de Sustentabilidad del Conacyt en México. Se agradece la información proporcionada por el CERTE del IIE.

Referencias

1. Andersen, S.K., Olesen, K.G., Jensen, F.V., Jensen, F.: Hugin: a shell for building bayesian belief universes for expert systems. In: Proc. Eleventh Joint Conference on Artificial Intelligence, IJCAI. pp. 1080–1085. Detroit, Michigan, U.S.A. (20-25 August 1989)
2. Chickering, D.: Optimal structure identification with greedy search. Journal of Machine Learning Research 3, 507–554 (2002)
3. Ibargüengoytia, P.H., Vadera, S., Sucar, L.: A probabilistic model for information and sensor validation. The Computer Journal 49(1), 113–126 (January 2006)
4. Neapolitan, R.: Learning Bayesian Networks. Prentice Hall, New Jersey (2004)
5. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, San Francisco, CA. (1988)
6. Xiang, Y.: A probabilistic framework for cooperative multi-agent distributed interpretation and optimization of communication. Artificial Intelligence 87, 295–342 (1996)

Propuesta de un sistema para optimizar el riego en invernaderos de plantas heterogéneas usando WNS y algoritmos evolutivos

Verónica del Rocío Ochoa López, Carlos Lino Ramírez, Rosario Baltazar Flores, Miguel Ángel Casillas Araiza, Víctor Manuel Zamudio Rodríguez, Sandra Jaqueline López Cervera, Guillermo Eduardo Méndez Zamora

Instituto Tecnológico de León,
División de Estudios de Posgrado e Investigación, León, Guanajuato, México

{rocio230483, carloslino, vic.zamudio}@itleon.edu.mx, {miguel.casillas, jclsandra, guillermomendez06}@gmail.com, r.baltazar@ieee.org

Resumen. En este trabajo se presenta la propuesta de un sistema de control de riego para lograr un uso eficiente de agua en invernaderos de plantas heterogéneas sin sacrificar el buen estado de las mismas, para ello se propone primero una clusterización (agrupamiento) de las plantas en base a las necesidades de riego y segundo la implementación de un algoritmo genético que establece los tiempos óptimos de riego y de espera que permitan minimizar la cantidad de agua utilizada al reducir el tiempo de riego y maximizar los tiempos de espera entre cada irrigación. Las distintas especies de plantas fueron agrupadas en base a su comportamiento de la humedad de la tierra bajo condiciones específicas, creando un vector de características que representa dicho comportamiento. El sistema permite un eficiente consumo del agua utilizada en los invernaderos.

Palabras clave: WSN, Xbee, Zigbee, optimización, agrupamiento, mejoramiento de invernadero.

Proposal of a System to Optimize Irrigation in Greenhouses of Heterogeneous Plants Using Evolutionary Algorithms and WNS

Abstract. In this paper, it is proposed an irrigation control system able to use in an efficient way the water in greenhouses of heterogeneous plants without sacrificing the good condition of them, to accomplish this, firstly it is proposed a clustering of plants based on the irrigation needs and secondly implementation of a genetic algorithm that establishes the optimum watering times and waiting between each watering which permit to minimize the amount of water used in each irrigation, minimizing watering time and maximizing the waiting times between each watering. Different species of plants were clustered based on their behavior in the soil moisture under specific conditions, for which a feature vector representing such behavior was created. The system allows efficient use of the water used in greenhouses

Keywords: WSN, Xbee, Zigbee, optimization, clustering, improving of greenhouse.

1. Introducción

De acuerdo a la Organización de las Naciones Unidas para la Alimentación, el riego representa el 70% de las extracciones de agua en el mundo, esto aunado al crecimiento de demográfico que para el año 2050 se espera aumente en un 40% representan una necesidad urgente de crear estrategias basadas en la ciencia y la tecnología para el desarrollo sostenible del uso de este recurso [1].

El desarrollo de sistemas de riego automatizados es una opción viable de contribuir al uso racional y eficiente del agua, ya que el diseño de estos sistemas busca suministrar sólo la cantidad de agua necesaria para la agricultura. Los sistemas de riego se pueden automatizar a través de la información del contenido volumétrico de agua del suelo, usando sensores de humedad para controlar los actuadores y ahorrar agua [2].

Un sistema de riego bien diseñado es un requisito esencial para un riego amigable rentable y ambiental. La tecnología de radiofrecuencia inalámbrica ha proporcionado oportunidades para implementar sistemas de comunicación inalámbrica de datos en la agricultura [3].

En los últimos años se han propuesto e implementado sistemas de apoyo a la horticultura de invernadero que permiten a los usuarios controlar el clima de los invernaderos de forma remota, transmitiendo información hasta los dispositivos como Smartphone o Tablet PC desde donde el usuario puede tomar decisiones o controlar ciertos dispositivos. Algunos de estos sistemas como [4, 5, 6, 7] integran micro-controladores y tecnología de bajo consumo de energía en el diseño de redes de sensores inalámbricas y proponen el uso de tecnología Zigbee en el monitoreo y control ambiental de invernaderos.

En este trabajo se propone un sistema que permite mejorar el riego en invernaderos de plantas de distintas especies y hacer un uso eficiente del agua, el modelo propone lo siguiente: el uso de pipetas para el suministro del agua, la clusterización de distintas especie de plantas en grupos que guarden una similitud en el comportamiento de la humedad de la tierra, la caracterización de los grupos obtenidos y por último la utilización de algoritmos metaheurísticos con el fin de obtener una solución que permita optimizar el consumo de agua al determinar el tiempo necesario de riego y maximizar en lo posible los tiempos de espera entre cada irrigación, sin afectar el crecimiento favorable de las plantas. El sistema trabaja con datos de humedad de la tierra en tiempo real por lo que se diseñó una arquitectura de red que permite obtener dicha información a través de nodos sensores y actuadores que trabajan bajo el protocolo de comunicación Zigbee.

2. Trabajos relacionados

Los grandes avances en el desarrollo y de WSN aplicados en el sector agrícola han hecho posible la automatización y mejoramiento del riego en invernaderos donde las condiciones ambientales pueden ser controladas de manera automática, tal es el caso

de trabajos como [4, 5 y 6] donde se implementaron sistemas de riego automatizados para optimizar el uso del agua para los cultivos agrícolas.

En [7] presenta el diseño de un software que da soporte a la toma de decisiones y su integración con una red de sensores inalámbricos (WSN) implementada en el campo en sitios específicos de control de riego por aspersión, el cual utiliza comunicación inalámbrica Bluetooth.

Los autores de [8] exploran el problema de despliegue óptimo de una WSN teniendo en cuenta la forma única de la disposición espacial del cultivo permanente durante el trasplante. En lugar de la agricultura convencional con fila horizontal o lineal se propuso un patrón hexagonal de la agricultura para los cultivos permanentes.

Otros trabajos como [9 y 10] buscan atacar el problema del desperdicio del agua utilizada en la producción agrícola, implementando sistemas de riego automatizados que utilizan un protocolo de comunicación Zigbee para la comunicación inalámbrica. [9] fue implementado en un invernadero de policultivo en el que cultivaron frijol, chile, maíz y tomate utilizando el método de irrigación por goteo. Mientas que en [10] calculan la cantidad de agua necesaria para las plantas.

También se han desarrollado investigaciones que trabajan con distintas metodologías para optimizar uso del agua. Tal es el caso de [11] donde los autores utilizan lógica difusa en conjunto con WSN como parte de su propuesta de un sistema de control de riego que normaliza el nivel de humedad deseado en el suelo agrícola mediante el control del flujo de agua de la bomba de riego, basado en las lecturas del sensor la bomba cambia los estados entre ON y OFF.

Los autores de [12] utilizan un algoritmo genético para la asimilación de datos y la gestión optima del agua. Además los autores de [13] utilizan técnicas de programación matemática para desarrollar una estrategia secuencial para la optimización del agua y la energía.

Por otra parte, la clasificación de las plantas es un tema que ha sido abordado en distintas investigaciones desde hace décadas, sin embargo, las primeras investigaciones y propuestas están basadas exclusivamente en la morfología de las plantas como se menciona en [14] donde se propone una clasificación de las plantas basada en cinco factores: 1) la silueta planta o forma general resultante de una combinación de otros sistemas, 2) el grupo de la hoja, 3) el tallo, 4) la raíz, y 5) la inflorescencia.

Con el surgimiento de técnicas de Inteligencia Artificial como el reconocimiento de patrones se han desarrollado investigaciones como [15] en la que se propone una clasificación de las plantas haciendo uso del algoritmo de agrupación k-means donde el vector de características considera los siguientes aspectos: constricción relativa de corola, longitud de la corola incluyendo lóbulos, presencia de glándulas en la corola, color de la corola, forma de cáliz lóbulos y presencia de glándulas en el cáliz.

3. Desarrollo de la propuesta

Se propone un modelo para mejorar el riego en invernaderos dedicados a la producción de diversas especies de plantas. Dicho modelo propone la utilización de pipetas para llevar a cabo el riego, la clusterización de las distintas especies en grupos

cuyo comportamiento de la humedad en la tierra es similar, seguido por una caracterización específica de cada grupo y por último la implementación de un algoritmo metaheurístico que permita establecer los tiempos óptimos de riego y espera para cada grupo definido. La implementación de este modelo en invernaderos de plantas heterogéneas permite hacer un uso eficiente del consumo de agua utilizada para el riego asegurando un crecimiento y estado favorable de las plantas.

3.1 Diseño de software y hardware del sistema

Se diseñó una arquitectura de red centralizada, la cual está conformada por: nodos sensores encargados de recabar información sobre variables de temperatura y humedad ambiental y de la tierra; nodos actuadores que permiten el flujo y suministro de agua a las plantas; un nodo inteligente que analiza la información obtenida y ejecuta un algoritmo de optimización que permite establecer los tiempos de riego y espera entre cada riego; un servidor que almacena los datos recabados por el sistema y una interfaz visual que accede a dichos datos para mostrarlos a manera de graficas al usuario. También cuenta con un nodo central encargado de la comunicación entre nodos y actuadores. Se utiliza la tecnología EEE 802.15.4 o Zigbee para la comunicación inalámbrica, ya que es capaz de soporta enrutamiento en malla permitiendo que los paquetes de datos atraviesen múltiples nodos (múltiple "saltos"), con el fin de investigar el nodo destino y haciendo posible que los nodos ZigBee se extiendan sobre grandes regiones y sigan apoyando la comunicación entre todos los dispositivos en la red [16].

En la figura 1 se puede observar la arquitectura propuesta.

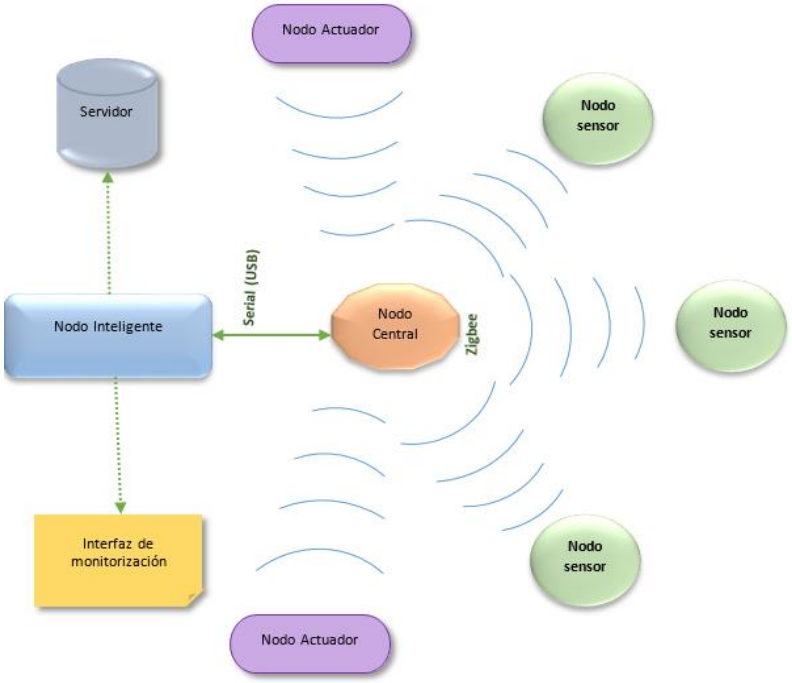


Fig. 1. Arquitectura propuesta.

Para la creación de los nodos sensores encargados de recabar información de las plantas y los nodos actuadores encargados del suministro del agua (Ver figura 2), se utilizaron sensores ambientales de temperatura y humedad, así como un sensor de humedad del suelo y una electroválvula solenoide respectivamente, los cuales están conectados a un microcontrolador y un módulo de comunicación inalámbrica Zigbee, con alimentación portátil (batería).

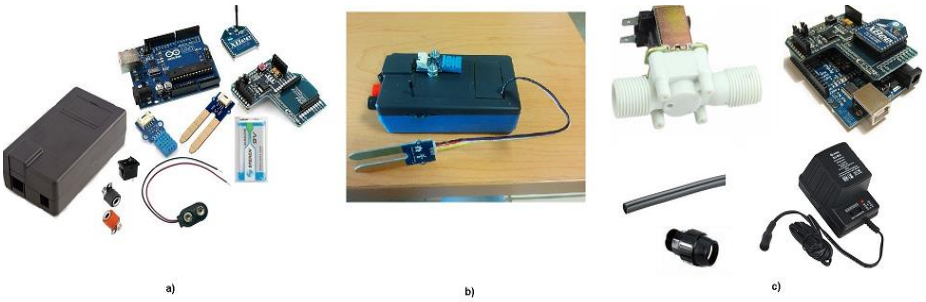


Fig. 2. a) Componentes del nodo sensor. **b**) Nodo terminado. **c)** Componentes del nodo actuador

3.2 Utilización de pipetas para el riego

Se propone el uso de pipetas para llevar a cabo el riego, ya que permiten un riego uniforme en todas las plantas del invernadero, asegurando el suministro de agua en cada una de ellas a diferencia del uso de regaderas o mangueras donde la humedad de las plantas después del riego no es uniforme

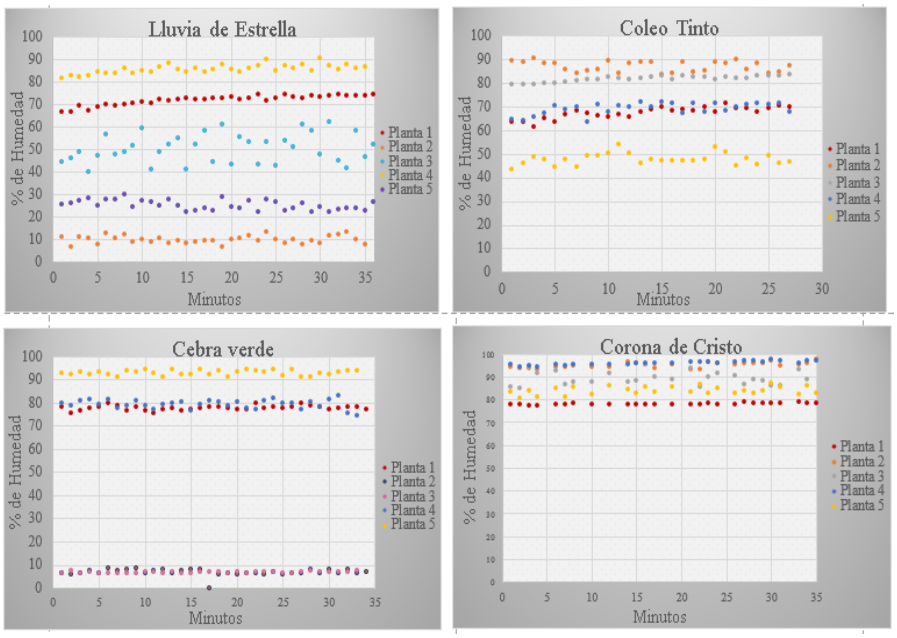


Fig. 3. Humedad de la tierra en las plantas usando el riego con regaderas.

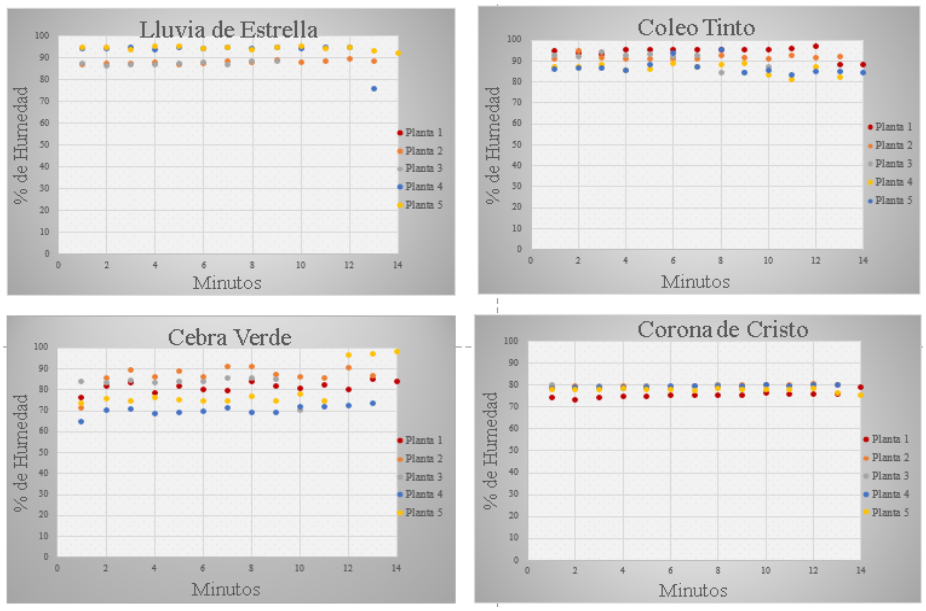


Fig. 4. Humedad de la tierra en plantas utilizando pipetas.

Las figuras 3 y 4 muestran 4 graficas de distintas especies de plantas donde se observa el nivel de humedad que obtuvo el sensor en un lapso de tiempo después de haber sido regadas haciendo uso de manguera o regadera (figura 4) y utilizando pipetas (figura 5). Con el uso de regadera no todas las plantas recibieron suficiente agua para lograr un nivel de humedad aceptable, mientras que con la utilización de pipetas todas las plantas recibieron agua logrando un nivel de humedad similar.

3.3 Clusterización de las plantas de acuerdo a las necesidades de riego

Se propone agrupar las distintas especies de plantas existentes en un invernadero buscando como característica común de cada grupo el comportamiento de la humedad de tierra, esto con el fin de obtener un número más reducido de grupos y hacer un mejor reacomodo de las plantas, al colocar en un misma hilera de pipetas todas aquellas que pertenecen a un mismo grupo y así poder ser regadas de la misma forma aprovechando el espacio y recursos del invernadero como las tomas de agua utilizada para el riego. El primer paso para poder formar los grupos es recabar información diaria sobre el comportamiento de la humedad de la tierra en las distintas especies, para ello se toma una muestra de cada especie, se riegan a un nivel máximo y se comienza un muestreo diario sobre el comportamiento de la humedad por 5 o más días si se desea, cuidando siempre de no llevar a las plantas a un estrés hídrico. En esta fase de hace uso de sensores de humedad del suelo, los cuales son colocados en las raíces de las plantas.

Generación de un vector de características

Una vez recolectada la información sobre la humedad de la tierra en las distintas especies de plantas se propone realizar un ajuste de datos con la información de cada

planta a fin de obtener una función que describa la curva formada por el comportamiento de la humedad en cada una de ellas. Esto permitirá que las plantas cuyas curvas sean semejantes en forma y posición formen parte de un mismo grupo. Se propone llevar a cabo distintas regresiones y seleccionar aquella función que mejor se ajuste a los datos. Los valores de los coeficientes de la función resultante son considerados como los elementos del vector de características. De tal forma que si se obtiene por ejemplo una función como la siguiente:

$$f(x) = p1 * x^2 + p2 * x + p3. \quad (1)$$

Los valores de *p1, p2 y p3* son considerados como el vector de características de la planta.

El grado de la función debe ser el mismo para todas las plantas para asegurar un vector de características del mismo tamaño. Esta propuesta para extraer características resulta simple y fue suficientemente efectiva en este estudio, ya que los vectores formados permitieron obtener grupos de plantas con necesidades de riego similar.

Creación de clusters y reacomodo de plantas

Una vez generado el vector de características para cada planta se utiliza el método de agrupación k-means, el cual agrupará las distintas especies en grupos cuyo comportamiento de la humedad de la tierra es similar entre ellos. Se propone el uso de k-means debido a que es uno de los algoritmos de aprendizaje no supervisado más conocido y cuya implementación requiere pocos recursos computacionales.

Después de obtener los grupos, las especies de plantas que hayan sido agrupadas en un mismo clúster, deben colocarse en una misma hilera de pipetas, la cual estará conectada a una válvula solenoide que permitirá el flujo del agua regando al mismo tiempo a cada una de las plantas pertenecientes a dicho grupo (ver figura 5).

3.4 Caracterización de los grupos formados

Se propone caracterizar cada uno de los grupos formados tomando en cuenta el comportamiento de las siguientes variables: humedad del suelo, volumen de tierra, altura de la maceta, posición del sensor, tiempo de riego (tiempo de activación de la válvula solenoide) y tiempo de espera para llegar a la humedad mínima. Se lleva a cabo un ajuste de datos del comportamiento de las variables obteniendo funciones matemáticas con distintos grados y seleccionando aquella cuyo error medio cuadrático sea menor.

Las funciones resultantes en esta etapa son utilizadas posteriormente como modelado del comportamiento del sistema en el algoritmo de optimización metaheurístico, con la finalidad de obtener tiempos óptimos de activación de la válvula y de espera para cada grupo.

3.5 Implementación de un modelo propuesto para optimizar el uso del agua

Se propone un modelo de optimización que permita ahorrar el consumo del agua a partir de la optimización de tiempos de riego y tiempos de espera entre cada riego

utilizando metaheurísticas y considerando variables como: humedad y temperatura ambiente y la humedad de las raíces de las plantaciones.

Fig. 5. Reacomodo de las distintas especies de plantas en los grupos definidos.

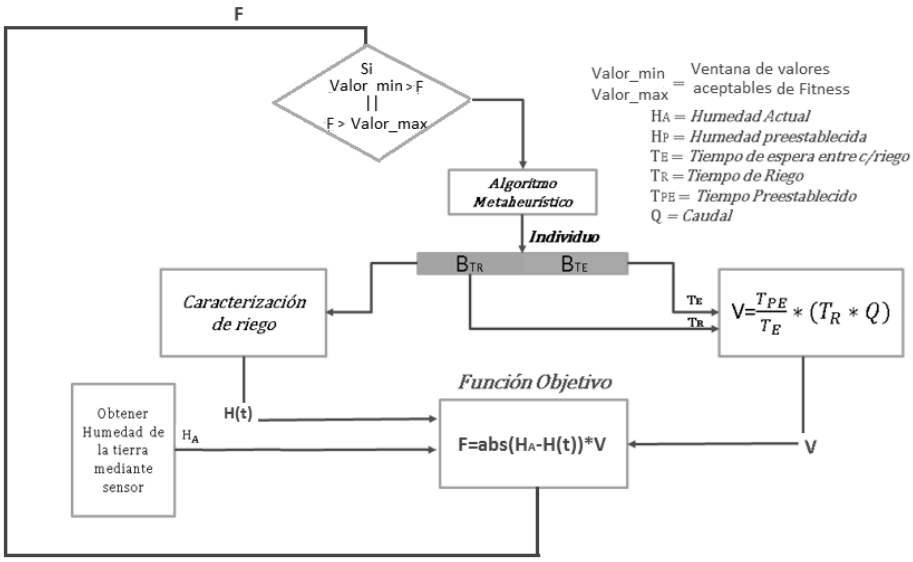


Fig. 6. Modelo de optimización propuesto.

El algoritmo metaheurístico funciona buscando el valor mínimo de la función objetivo (fitness), cuando el valor del fitness se encuentra entre dos números preestablecidos (a la que llamamos ventana del fitness deseada) entonces el algoritmo se detiene y se ajustan los tiempos (de riego y de espera) en las electroválvulas para que se mantengan accionando.

El sistema continuamente toma muestras de la humedad en las raíces para compararlo con la curva de humedad calculada a partir de los tiempos obtenidos por el algoritmo. Si existe una diferencia considerable, entonces el fitness tiene un valor fuera de la ventana del fitness deseada, por lo que el algoritmo volverá a calcular valores de tiempos considerando que pudieron haber cambiado los factores externos como temperatura y humedad ambiental que afectaron el comportamiento de humedad en la tierra.

El proceso seguido por el modelo se muestra en la figura 6.

Descripción del modelo

Para lograr el consumo de agua mínimo, debe considerarse que se busca el mínimo de tiempo de riego y el máximo tiempo de espera entre cada riego, además de buscar la máxima humedad posible permitida en las plantas.

A continuación se describe el modelo diseñado:

- El algoritmo metaheurístico comienza a optimizar tiempos
- Se obtiene el nivel de humedad actual de un grupo de plantas mediante un sensor de humedad del suelo.
- Si la diferencia de la humedad de la tierra actual con la humedad calculada por el algoritmo se encuentra dentro de la ventana establecida de nivel aceptado, entonces el algoritmo se detendrá.
- La optimización (por medio del algoritmo metaheurístico) se realiza a partir de la caracterización de las plantas de cada grupo definido. Con la caracterización se puede obtener a partir de un tiempo definido de riego (T_R) y la humedad del suelo censada (H_A), una humedad promedio que presentaría la planta (H_t). Por otra parte, para la determinación del volumen consumido de agua dados los tiempos de riego (T_R y T_E) se pueden establecer las siguientes ecuaciones:

$$V = T_R * \mathrm{Q}, \tag{2}$$

$$f = \frac{T_{PE}}{T_E}, \tag{3}$$

$$V = f * (Vtr), \tag{4}$$

dónde: Vtr = Volumen consumido en tiempo de riego, T_R =tiempo de riego, Q= caudal de la electroválvula, T_{PE} = tiempo preestablecido (una semana), T_E=tiempo de espera entre cada riego y f= frecuencia de riego, V= volumen utilizado durante un lapso de tiempo T_{PE}.

- Se propone la utilización de algoritmos evolutivos para la optimización, en este caso un algoritmo genético. Se pretende que cada individuo esté conformado por una cadena de bits, donde la mitad de dicha cadena representa el tiempo de espera y la otra mitad el tiempo de riego. De tal forma, que el algoritmo metaheurístico genera los individuos mediante un motor pseudoaleatorio, estos individuos se prueban mediante la función objetivo representada por la siguiente ecuación:

$$F = abs(H_A - Ht) * V. \tag{5}$$

Así lo que se pretende es minimizar la función objetivo.

- La solución obtenida por el algoritmo será utilizada para ajustar los tiempos que la electroválvula permanecerá activa permitiendo el flujo del agua y alcanzar la humedad deseada para las plantas, así como el tiempo que deberá permanecer desactivada hasta el próximo riego.

4. Experimentación y resultados

Para realizar la experimentación se tomó una muestra de 48 plantas pertenecientes a 13 especies distintas, esta muestra incluía desde plantas de riego frecuente hasta plantas del grupo de las cactáceas o suculentas, las cuales son capaces de almacenar agua en sus hojas y tallos y por lo tanto su riego es más esporádico.

Se hizo un muestreo por 6 días sobre el comportamiento de la humedad de las plantas pertenecientes a la muestra, posteriormente se llevó a cabo un ajuste de datos con la información recabada de cada planta y se obtuvieron varias funciones de distintos grados que describían el comportamiento de la humedad en cada una de ellas. La función seleccionada debido a un mejor ajuste en los datos de cada planta y cuyo vector resultante representó un menor costo computacional al utilizarse en el algoritmo de agrupación fue un polinomio de segundo grado conformado por 3 coeficientes, cada uno de los cuales fue considerado como una característica para el vector de cada planta. Una vez que se obtuvieron los vectores de características de 429 instancias se procedió a realizar el agrupamiento de las distintas especies, para ello se utilizó el método de agrupamiento k-means. El algoritmo fue ejecutado en varias ocasiones asignando distintos valores de K, desde K=2 hasta K=13 (número máximo de especies de plantas), obteniendo distintos grupos en cada una de dichas ejecuciones.

Para evaluar los grupos formados en cada prueba y seleccionar las agrupaciones más adecuadas se tomó como criterio la desviación estándar de la distribución de las especies en cada grupo. En la tabla 1 puede observarse que la desviación estándar obtenida de los grupos formados con K=4 fue menor que el resto, por lo que éstos grupos fueron considerados para llevar a cabo el reacomodo y optimización de los tiempos de riego.

Tabla 1. Grupos formados con distintos valores de K.

Valor de K	Distribución de especies en cada grupo	Desviación estándar de la distribución
2	11,2	7.071067812
3	6, 6, 1	2.886751346
4	**4,4,3,2**	**0.957427108**
5	7, 1, 3, 2	2.62995564
6	2,4,2,1,1,3	1.16904519
10	4, 1, 2, 1, 1, 1, 2, 1	1.060660172

Los 4 grupos formados se muestran en la tabla 2 y de manera gráfica pueden visualizarse en la figura 7.

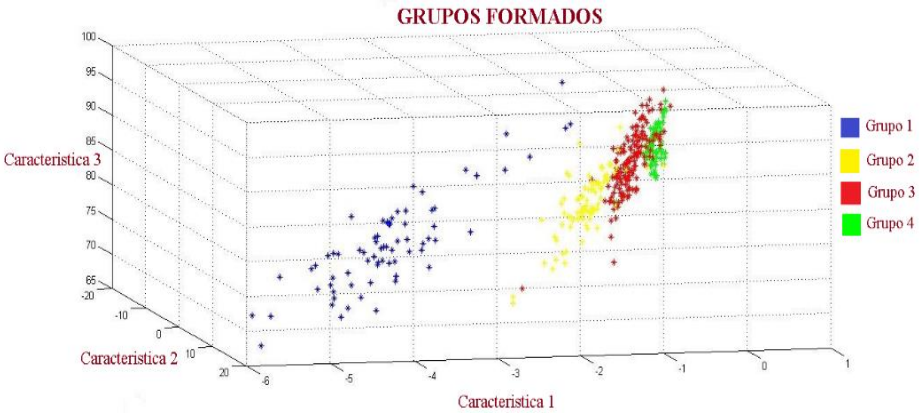


Fig. 7. Grupos formados utilizando el vector de características propuesto.

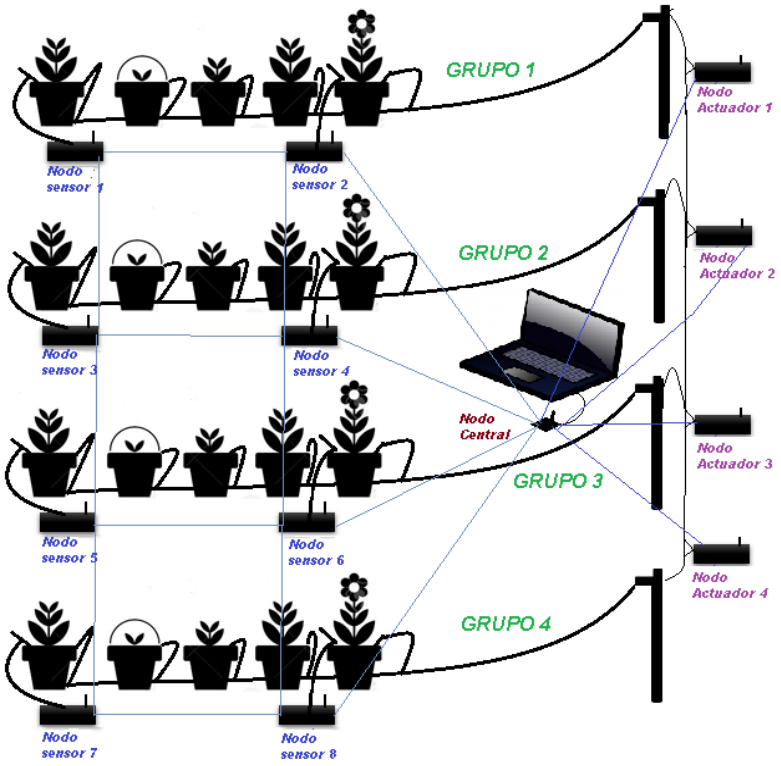


Fig. 8. Reacomodo de las plantas y distribución de los sensores y actuadores.

Con los resultados de la clusterización se hizo un reacomodo de las plantas en base a los grupos formados como se muestra en la figura 8. De este modo al implementar el sistema de riego automatizado las plantas cuyo ciclo de riego es similar estarán acomodadas en una misma fila y se les suministrará de agua al mismo tiempo.

Los sensores fueron configurados para enviar información al nodo central sobre el porcentaje de humedad de la tierra, además de enviar el número del grupo al que pertenecen (1, 2, 3 o 4); el nodo central se encarga de recibir la información y activar las electroválvulas de cada grupo de acuerdo a los tiempos propuestos por el algoritmo de optimización.

Tabla 2. Grupos formados de acuerdo al ciclo de riego.

Grupo 1	Grupo 2	Grupo 3	Grupo 4
Cebra Verde	Corona de Cristo	Lluvia	Coleo Tinto
Enredadera	Estrellita	Pata Elefante	Lavanda
Estrella	Helecho	Real de Oro	
Rosita	Santolín		

Para la caracterización de cada grupo se llevaron a cabo algunas pruebas que permitieron describir mediante funciones el comportamiento de la humedad en la tierra en cada grupo durante y después del riego, para dichas pruebas se utilizaron plantas con distinto volumen de tierra, plantas en macetas con distintas alturas, además de considerar distintas posiciones para la colocación de los sensores. La utilización de distintos valores de las variables mencionadas permite que el sistema calcule los tiempos de riego y espera para distintos tamaños de plantas.

Las funciones obtenidas en esta etapa se describen a continuación:

1. *Función tiempo de riego vs humedad alcanzada:* describe el comportamiento de la humedad con respecto al tiempo en segundos que se mantuvo activa la electroválvula.
2. *Función tiempo de riego vs volumen:* describe el tiempo en segundos que fue necesario mantener activa la electroválvula para alcanzar una humedad deseada en macetas con distintos volúmenes de tierra.
3. *Función tiempo de riego vs altura:* describe el tiempo en segundos que fue necesario mantener activa la electroválvula para alcanzar una humedad deseada en macetas con distintas alturas.
4. *Función tiempo de riego vs posición del sensor:* describe el tiempo en segundos que se mantuvo activa la electroválvula para alcanzar una humedad deseada en diferentes posiciones de la maceta.
5. *Función tiempo de espera vs humedad alcanzada:* describe el tiempo en segundos transcurrido después del riego hasta llegar a la humedad mínima establecida para el grupo.
6. *Función tiempo de espera vs volumen:* describe el tiempo en segundos transcurrido después del riego hasta llegar a la humedad mínima en macetas con distintos volúmenes de tierra.
7. *Función tiempo de espera vs altura:* describe el tiempo en segundos transcurrido después del riego hasta llegar a la humedad mínima en con distintas alturas.
8. *Función tiempo de espera vs posición del sensor:* describe el tiempo en segundos transcurrido después del riego hasta llegar a la humedad mínima en diferentes posiciones de la maceta.

En la tabla 3 se muestran las funciones obtenidas para el grupo 1. No obstante, para el resto de los grupos se obtuvieron sus ecuaciones respectivas, las cuales fueron utilizadas en la fase de optimización.

Tabla 3. Funciones obtenidas para la caracterización del grupo 1.

Relación	Función	Bondad de ajuste	
Tiempo de Riego vs Humedad	45.3-39.7*cos(x*0.1256)-2.022*sin(x*w)	SSE R-square Adjusted	521.8 0.9799
Tiempo de Riego vs Volumen de tierra	-1.042*x^2 +157.2*x - 4550	SSE R-square Adjusted	8.4E-23 1
Tiempo de Riego vs Altura de maceta	-0.0074*x^2 +1.034*x -24.24	SSE R-square Adjusted	2.3E-27 1
Tiempo de Riego vs Posición del sensor	0.01894*x^2 -2.217*x + 65.59	SSE R-square Adjusted	2.4E-27 1
Tiempo de Espera vs Humedad	2.86e-10*x^2-0002842*x+ 78.95	SSE R-square Adjusted	1.8E-27 1
Tiempo de Espera vs Volumen de tierra	-1.33e-06*x^2 +0.2595*x-1.15e+04	SSE R-square Adjusted	3.6E-23 1
Tiempo de Espera vs Altura de la maceta	-1.40e-09*x^2 +0.000247*x-1.791	SSE R-square Adjusted	2.6E-28 1
Tiempo de Espera vs Posición del sensor	973.4*x^2 -2.63e+04*x+2.04e+05	SSE R-square Adjusted	5.1E-21 1

Por último para la etapa de optimización se ejecutó el Algoritmo Genético (AG) [17 y 18], utilizando los siguientes parámetros: Elitismo=10%, Selección= Ruleta pesada, Cruza= En un punto, Muta=One-flip.

Los resultados obtenidos durante la experimentación fueron satifactorios ya que los tiempos de riego y espera permitieron mejorar el consumo de agua con respecto al riego convencional en cada uno de los grupos definidos, sin sacrificar el buen estado y crecimeitno favorable de las plantas.

5. Conclusiones y trabajo a futuro

En este trabajo se abordó la problemática del uso eficiente del agua en invernaderos de plantas heterogéneas o con distintas necesidades de riego, presentando un modelo en el que se propone la agrupación de las distintas especies de plantas en grupos cuyas

necesidades de riego sean similares y el uso de un algoritmo genético que permite determinar la cantidad de agua necesaria para cada grupo definido.

Se puede observar de acuerdo a los resultados obtenidos que la propuesta para generar un vector de características basado en los coeficientes de la ecuación que describe el comportamiento de la humedad de la tierra fue suficientemente efectivo para hacer una distribución de las plantas en grupos más homogéneos (con necesidades de riego similares).

Las funciones obtenidas en la fase de caracterización en conjunto con un algoritmo genético permitieron determinar los tiempos de riego y espera que hicieron posible minimizar la cantidad de agua utilizada para el riego, sin sacrificar el crecimiento favorable de las plantas. La implementación del algoritmo meta heurístico permite buscar una solución que puede ser óptima para el consumo de agua, gracias a la implementación de esta estrategia de inteligencia artificial nos permitirá en trabajo futuro hacer estudios enfocados a optimización preventiva en base a modelos de un invernadero, donde se buscará no solo optimizar el consumo de agua sino cambiar las formas de las curvas en la caracterización para adaptarlas al ambiente real con diferentes perturbaciones y lograr un control dinámico. Para resolver esto, el espacio de búsqueda aumentará considerablemente, es aquí donde se podrá observar la potencialidad de la estrategia propuesta utilizando algoritmos heurísticos.

El uso de tecnologías como las redes de sensores inalámbricas permitió llevar a cabo el muestro y monitoreo de las variables que ayudaron a obtener la caracterización de cada grupo definido, además de permitir el riego automatizado de las plantas.

El sistema permitirá realizar estudios posteriores para realizar investigación multiobjetivo, además de recolección de datos para realizar minería de datos y avances en el paradigma de *Internet of Things*.

Agradecimientos. El presente trabajo se desarrolló con apoyo del Consejo Nacional de Ciencia y Tecnología de México, a través del Programa de Becas Nacionales para estudios de posgrado (Beca 387098), del Tecnológico Nacional de México (Instituto Tecnológico de León), del CONCYTEG y del Instituto de la Memoria.

Referencias

1. Programa de ONU. Agua para la promoción y la comunicación en el marco del decenio (UNW-DPAC). Agua y agricultura en la economía verde (2015)
2. Nemali, K.S., Van Iersel, M.W.: An automated system for controlling drought stress and irrigation in potted plants. Sci. Horticult, Vol. 110, No. 3, pp. 292–297 (2006)
3. Oksanen, T., Ohman, M., Miettinen, M., Visala, A.: Open configurable control system for precision farming. ASABE St. Joseph MI. (2004)
4. Gutiérrez A., J., Villa Medina, J.F., Nieto Garibay, A., Porta Gándara, M.A.: Automated irrigation system using a wireless sensor network and gprs module. IEEE Transactions on instrumentation and measurement, Vol. 63 (2014)
5. Li, X-H., Cheng, X., Yan, K., Gong, P.: A monitoring system for vegetable greenhouses based on wireless sensor network. Sensors (2010)
6. Zhang, Q. Yang, X-L., Zhou, Y-M., Wang, L-R., Guo, X-S.: A wireless solution for greenhouse monitoring and control system based on ZigBee technology. Journal of Zhejiang University SCIENCE A ISSN (2007)

7. Hema, N., Kant K.: Optimization of sensor deployment in WSN for precision irrigation using spatial arrangement of permanent crop. IEEE (2013)
8. Kim, Y., Evans, R.G.: Software design for wireless sensor-based site-specific irrigation. Computers and Electronics in Agriculture (2009)
9. Torres Lopes, J.G.: Diseño de un sistema de irrigación automatizado para poli-cultivo. Universidad Autónoma de Querétaro (2012)
10. Tao Chi, Ming Chen, Qiang Gao: Implementation and study of a greenhouse environment surveillance system based on wireless sensor network. The 2008 International Conference on Embedded Software and Systems Symposia (ICESS 2008) (2008)
11. Castro Popoca, M., Águila Marín, F.M., Quevedo Nolasco, A., Kleisinger, S., Tijerina Chávez, L., Mejía Sáenz, E.: Sistema de riego automatizado en tiempo real con balance hídrico, medición de humedad del suelo y lisímetro (2008)
12. Amor V.M., I., Honda, K., Das Gupta, A., Droogers, P., Clemente, R.S.: Combining remote sensing-simulation modeling and genetic algorithm optimization to explore water management options in irrigated agriculture. Agricultural Water Management (2006)
13. Grossmann, I.E., Martin, M.: Energy and water optimization in biofuel plants. Chinese Journal of Chemical Engineering (2010)
14. Halloy, S.A.: Morphological classification of plants, with special reference to the New Zealand alpine flora. Journal of Vegetation Science 1, pp. 291–304 (1990)
15. Orloci, L.: Automatic classification of plants based on information content. Canadian Journal of Botany (2011)
16. XBee®/XBee-PRO® ZB RF. Modules. Digi International Inc (2012)
17. Jones, M.T.: Artificial Intelligence A System Approach. Infinity Science Press LLC Hingham, Massachusetts New Delhi (2008)
18. Talb, El G.: Metaheuristics from design to implementation. University of Lille CNRS INRIA by John Wiley and Sons, Inc. (2009)

Modelado y análisis formal de jugadas del fútbol

Jonathan Tellez-Giron, Matías Alvarado

Centro de Investigación y de Estudios Avanzados del IPN, CDMX, México

Resumen. Desarrollamos un modelado analítico del fútbol soccer, y se proponen fórmulas para medir la *ocurrencia promedio de jugadas por minuto* y *la eficiencia de un jugador* en cada posición de juego. Con datos de la Liga Española sobre la ocurrencia de jugadas por jugador, se validan las formulas propuestas. La secuencia correcta de jugadas se logra mediante la función de transición de un autómata finito para el futbol.

Palabras clave: Modelo de fútbol soccer, autómatas concurrentes, eficiencia de jugador.

Modeling and Formal Analysis of Football Plays

Abstract. We present an analytical modeling of soccer football, and formulas to measure the *average occurrence of plays per minute* and *the efficiency of a player* in each playing position are proposed . Using data from the Spanish League on the occurrence of plays per player, we validate the proposed formulas. The correct sequence of moves is by means of the transition function of a finite automaton to soccer football.

Keywords: Soccer football modeling, concurrent finite automaton, player's efficiency.

1. Introducción

El fútbol soccer (FS), del ingles *football* o *balompié*, es uno de los juegos más populares a nivel mundial, inventado en la gran Bretaña desde la Edad Media y trasladado fuera por los marineros británicos. A mediados del siglo *XVII* estudiantes de la universidad de Cambridge desarrollaron las primeras reglas oficiales para este deporte [3]. El fútbol se juega en una cancha de césped de 120 x 60 m, con arcos o *porterías* en cada lado del campo, entre dos equipos de 11 jugadores cada uno. Un partido de fútbol inicia con el balón a media cancha; un jugador del equipo que ganó el volado para dar el primer toque al balón lo pasa a un compañero. Sobre el césped los jugadores deben conducir el balón con los pies y pueden controlarlo, asimismo, con piernas, cuerpo y cabeza, pero no con brazos ni manos, salvo para los saques de banda. A base de conducción

individual y pases del balón entre compañeros y de esquivar a los rivales, deben buscar introducir el balón en la portería rival para anotar goles; gana el partido el equipo que anota mayor número de goles. Los roles de juego, acorde a su posición en la cancha respecto a la portería propia son delanteros, medios, defensas, y el portero, siendo la misión del portero evitar anotaciones de gol en su portería, y sólo él tiene permitido el manejo del balón con manos y brazos.

En el FS, como en todo juego, los jugadores pretende realizar la mejor acción posible en beneficio del equipo, con base en elegir la mejor estrategia considerando las condiciones del juego y, especialmente, las estrategias del resto de los jugadores. La selección de estrategias es un aspecto esencial a considerar en la automatización del FS. Actualmente, el modelado formal y el análisis estratégico para juegos multi-jugador como el fútbol americano o el béisbol [2,4,1], es relevante en la ciencia de los deportes, la computación y la teoría de juegos.

A continuación se presentan las jugadas del FS, haciendo mención de las jugadas de cada jugador, clasificadas de acuerdo a su posición, en la Tabla 1: delanteros, mediocampistas, defensas y portero. El conjunto de jugadas en la Tabla 1, se tomaron de la página web oficial *Liga BBVA* y la (`http://mex.laliga.es/liga-bbva`), donde se muestra los datos de cada equipo, los datos relevantes por jugador, así como el conteo de jugadas relevantes. La página web se dividen en cinco grandes categorías: *jugadas defensivas, jugadas de construcción, jugadas de ataque, jugadas de gol* y por último una categoría que refleja la *disciplina* de los jugadores.

2. Ocurrencia de jugadas y eficiencia de jugadores

2.1. Frecuencia de jugadas

Las jugadas para las posiciones de delantero, mediocampista y defensa son las mismas, porque para ellos no existen reglas que limiten su movilidad en el campo. La diferencia por rol es la probabilidad de que suceda una jugada especifica: por ejemplo, un delantero puede hacer una barrida para quitar el balón sin embargo la probabilidad de que un delantero promedio haga una recuperación (por minuto) es de 4.62 % mientras que de un defensa es de 6.44 %; inversamente, podemos observar que la probabilidad de que un defensa anote gol es de 0.05 % mientras que la probabilidad de que un delantero anote gol es de 0.99 %. El portero tiene jugadas propias, dada su función de evitar que el equipo contrario anote gol y su posibilidad de utilizar las manos para lograr su objetivo.

Para el análisis se toman en cuenta el número de ocurrencias de las jugadas para los jugadores mas populares de cada equipo, así como a los equipos mejor clasificados conforme los puntos obtenidos durante el torneo; asimismo, la diferencia entre goles anotados y recibidos. En caso de empate de puntos entre tres o más equipos se considera la misma diferencia en los partidos que involucren a dichos equipos, y los goles anotados en total a lo largo de la liga. Los datos son de la primera jornada en agosto 2015 hasta la jornada 30 del torneo en marzo 2016.

Tabla 1. Clasificación de Jugadas en el FS.

Jugadas defensivas

Jugada	*Descripción*	*Jugada*	*Descripción*	*Jugada*	*Descripción*
db^i	bloquear disparo	r^i	recuperación	i^i	intercepción
d^i	despejar	$blce^i$	bloqueo con éxito	$blse^i$	bloqueo sin éxito

Jugadas de construcción

Jugada	*Descripción*	*Jugada*	*Descripción*	*Jugada*	*Descripción*
pl^j	pase largo	pc^i	pase corto	ce^i	centro

Jugadas de ataque

Jugada	*Descripción*	*Jugada*	*Descripción*	*Jugada*	*Descripción*
t^i	tiro	as^i	asistencia	rea^i	regate acertado
ref^i	regate fallido				

Jugada de gol

Jugada	*Descripción*
go^i	*gol*

Disciplina

Jugada	*Descripción*	*Jugada*	*Descripción*	*Jugada*	*Descripción*
fap^i	falta propia	rep^i	recibir penalti	fuj^i	fuera de juego
far^i	falta recibida	pec^i	cometer penalti	per^i	perder balón
mac^i	cometer mano	adv^i	advertencia	exp^i	expulsión
sdu^i	sanción débil uno	sdd^i	sanción débil dos	sf^i	sanción fuerte
sa^i	sanción acumulada				

Jugadas de Portero

Jugada	*Descripción*	*Jugada*	*Descripción*	*Jugada*	*Descripción*
pd^i	parar disparo	rd^i	rechazar disparo	sp^i	sacar y pasar balón

Con base en dichos datos se hace un análisis estadístico para obtener la ocurrencia de las jugadas por jugador. El tiempo de juego por jugador es variable debido a posibles cambios y/o expulsiones a lo largo de los encuentros. Así, la cantidad de minutos jugados puede variar incluso teniendo la misma cantidad de partidos jugados. Estos datos se normalizaron obteniendo la cantidad promedio de jugadas por partido y por minutos jugados para cada uno de los jugadores. Esto permite obtener una ocurrencia promedio por minuto jugado, cuyos valores se muestran en la Tabla 2.

2.2. Eficiencia de un jugador

A partir de la *ocurrencia total de jugadas en la Liga* calculamos:

- La *ocurrencia promedio de jugadas por partido*, ecuación 1.
- La *ocurrencia promedio de jugadas por minuto*, ecuación 2.

$$OPP = \frac{\sum_{i=1}^{N} \frac{j(i)}{p(i)}}{N}, \tag{1}$$

$$OPM = \frac{\sum_{i=1}^{N} \frac{j(i)}{m(i)}}{N}, \tag{2}$$

donde, OPP = ocurrencia promedio de jugadas por partido, OPM = ocurrencia promedio de jugadas por minuto, N = número de jugadores, $j(i)$ = número de jugadas en la Liga por el jugador i, $p(i)$ = número de partidos jugados por el jugador i, $m(i)$ = número de minutos jugados por el jugador i. Estos datos se utilizan para calcular la eficiencia de cada jugador. De forma analítica, la OPM permite mayor precisión y evita ambigüedad en cuanto a la eficiencia de un jugador. Por ejemplo, si un jugador esta presente pocos minutos de un partido, y en pocos minutos es capaz de realizar una anotación, entonces dicho jugador es muy eficiente. Por tanto nuestro interés se centra en la *eficiencia por minutos jugados* (**emj**).

Tabla 2. Ocurrencia promedio por minutos jugados para delanteros en Liga BBVA 2015/2016.

Jugada	*Ocurrencia*	*Probabilidad*	*Jugada*	*Ocurrencia*	*Probabilidad*
d^i	0.0068	0.00013104	rep^i	0.0000	0.00000000
pl^j	0.0168	0.02439200	rea^i	0.0106	0.00020360
pc^i	0.3349	0.48752592	go^i	0.0068	0.00991233
ce^i	0.0130	0.01893542	fap^i	0.0151	0.02203089
t^i	0.0296	0.04312250	far^i	0.0184	0.02680279
as^i	0.0028	0.00005410	adv^i	0.0130	0.01888120
ref^i	0.0130	0.00025002	sdu^i	0.0021	0.00308974
r^i	0.0317	0.00060920	sdd^i	0.0000	0.00005995
i^i	0.0000	0.00000000	sf^i	0.0000	0.00000000
fuj^i	0.0099	0.01446414	$blce^i$	0.0074	0.00014196
$blse^i$	0.0028	0.00005383	per^i	*Determinado*	
pec^i	0.0000	0.00005994	sa^i	*por otras*	
mac^i	0.0011	0.00165763	exp^i	*jugadas*	
db^i	0.0055	0.00010564			

El análisis de **emj** para los jugadores tuvo el objetivo de diferenciar las características principales de los diferentes roles y su desempeño en cada una de ellas. Los criterios de evaluación de **emj** son : *bloqueos*, *duelos cuerpo a cuerpo*, *duelos aéreos*, *pases largos*, *pases cortos*, *centros*, *regates*, *penaltis* y *goles*. La **emj** de los criterios mencionados se calcula a partir de la siguiente formula:

$$\textbf{emj} = \frac{Ocurrencia\ promedio\ de\ jugadas\ exitosas\ por\ minutosjugados}{Ocurrencia\ promedio\ de\ jugadas\ totales\ por\ minutosjugados}$$

Por ejemplo, en el caso de los *goles* se calcula como

$$\frac{Goles\ promedio\ por\ minutosjugados}{Cantidad\ de\ tiros\ promedio\ por\ minutosjugados}.$$

Con este análisis se obtuvieron resultados tales que permiten determinar, claramente, la **emj** de cada rol. Los resultados se muestran en la Tabla 3. El comparativo de eficiencia entre jugadores de la Liga BBVA en la Tabla 4. La **emj** de un *delantero promedio* es un indicador de contraste respecto a la emj del resto de los jugadores: la eficiencia de *Lionel Messi* en pases cortos (82.6 %) sobre el promedio (73.39 %) identifica a este jugador como *coordinador de juego* del equipo; a diferencia, el porcentaje de *Luis Suarez* en goles es 27.08 % sobre el promedio (22.53 %), indicando que su rol en el juego es de *caza-goles*. De este análisis puede verse cómo, a pesar de que ambos jugadores pertenecen al mismo equipo y tienen el mismo rol como delanteros, su función individual se caracteriza acorde a su eficiencia en determinados aspectos del juego. La *habilidad* del jugador es factor primordial durante el desarrollo de un juego, caracteriza su rol en el campo e indica una preferencia sobre sus *elecciones estratégicas*.

Tabla 3. Eficiencia promedio por minutos jugados (%) en la Liga BBVA 2015/2016.

Jugador Promedio	Bloqueos	Duelos cuerpo a cuerpo	Duelos aéreos	Pases largos	Pases cortos	Centros	Tiros a puerta	Regates	Penaltis	Goles
Delantero	72.62	42.85	38.39	57.06	73.39	19.58	55.25	45.49	23.75	22.53
Medio	71.28	52.60	45.23	62.55	83.13	25.99	40.58	56.97	5.26	8.38
Defensa	71.28	56.81	57.65	45.96	86.39	15.78	34.98	58.22	0.00	7.39

Tabla 4. Eficiencia de jugadores por minutos jugados (%) en la Liga BBVA 2015/2016.

Jugador	Bloqueos	Duelos cuerpo a cuerpo	Duelos aéreos	Pases largos	Pases cortos	Centros	Tiros a puerta	Regates	Penaltis	Goles
Luis Suárez	82.35	43.4	46.43	48.48	72.95	6.9	55.21	35.56	33.33	27.08
Lionel Messi	62.5	54.95	25	72.41	82.6	21.43	63.33	65.33	42.86	24.44
Neymar	61.9	54.86	43.33	74.63	81.52	30.77	56.04	58.08	66.67	23.08
Griezmann	76	47.01	33.1	78.18	79.01	36.84	55.93	48.48	50	28.81
Fernando Torres	75	38.93	49.5	25	67.52	40	47.83	42.11	0	21.74
Cristiano Ronaldo	80	46.24	56.04	60	80.04	16.28	52.5	52.05	66.67	17.5
Karim Benzema	75	41.84	28.57	85.71	81.93	5.56	64.62	45.71	0	30.77
Gareth Bale	75	53.7	30	73.33	80.24	23.53	50	58.18	0	27.78
Soldado Rillo	55	49.71	24.68	52.17	69.22	9.52	54.55	55	0	12.12
Cedric Bakambu	80	34.59	25.53	63.64	73.5	15	65	58.62	0	27.5

En resumen, el análisis de los datos estadísticos o de frecuencia de ocurrencia de jugadas del FS, expresa de manera realista, lo que cada jugador puede realizar según las condiciones de ocurrencia de las jugadas.

3. Lenguaje formal y autómata finito para el fútbol

El estudio previo sobre la frecuencia de ocurrencia de jugadas, permite obtener *fotografías* de la eficiencia de los jugadores en cierto momento de un torneo. Para modelar en general la dinámica del FS y las combinaciones que pueden ocurrir, es necesario otra herramienta de carácter algorítmico, tal que permita la simulación dinámica de cualquier partido posible. Se construyó una *Gramática Libre de Contexto* y un Autómata Finito (AF) para tal fin. Los estados principales son: cuando un jugador está en *posesión* y *sin posesión* del balón. A partir de estos estados se modela el FS para todo jugador, con excepción del portero. El conjunto de estados del jugador se presenta en la Tabla 5.

Tabla 5. Conjunto de estados de un jugador en el fútbol.

Estado	*Descripción*	*Estado*	*Descripción*
P	Posesión del balón	X	Estado posterior a expulsión
SP	Sin posesión del balón	FJ	Estado posterior a fuera de juego
G	Estado posterior a un gol	TA	Tarjeta amarilla
F	Estado posterior a una falta	STA	Segunda tarjeta amarilla
TR	Tarjeta Roja		

Las transiciones entre los estados de juego son a través de las jugadas posibles en cada estado. La *función de transición* del AF se define como $\delta : Q \times \Sigma \to Q$, donde Q es el conjunto finito de estados y Σ es el alfabeto, en este caso de las jugadas de FS. Así, la función define las transiciones de un estado hacia otro, a través de las jugadas que se dan en la Tabla 6.

3.1. Construcción de jugadas

El AF, ilustrado en la Figura 1, es definido como sigue:

- $Q = \{P, SP, G, F, TR, X, FJ, TA, STA\}$,
- $\Sigma = \{d^i, pl^j, pc^i, ce^i, t^i, as^i, ref^i, r^i, i^i, exp^i, rep^i, rea^i, go^i, fap^i, far^i, adv^i, sdu^i, sdd^i, sf^i, sa^i, fuj^i, per^i, db^i, pec^i, mac^i, sp^i, pd^i, rd^i, blce^i, blse^i\}$,
- δ : función de transición, definida en Tabla 6,
- $q_0 = P$,
- $F = \{G, X\}$.

Cada transición del AF corresponde a una regla de la GLC -la cual hemos omitido. Se diseño un algoritmo con base en el AF definido previamente, que se

Tabla 6. Transiciones entre estados de un jugador en el fútbol.

Función	*Transición*	*Función*	*Transición*	*Función*	*Transición*
$\delta(P,\ far^i)$	P	$\delta(P,\ t^i)$	SP	$\delta(SP,\ i^i)$	P
$\delta(P,\ rep^i)$	P	$\delta(P,\ as^i)$	SP	$\delta(SP,\ blce^i)$	P
$\delta(P,\ rea^i)$	P	$\delta(P,\ ref^i)$	SP	$\delta(FJ,\ per^i)$	SP
$\delta(P,\ go^i)$	G	$\delta(SP,\ far^i)$	SP	$\delta(F,\ sdv^i)$	SP
$\delta(P,\ fap^i)$	F	$\delta(SP,\ blse^i)$	SP	$\delta(F,\ sdu^i)$	TA
$\delta(P,\ fuj^i)$	FJ	$\delta(SP,\ db^i)$	SP	$\delta(F,\ sdd^i)$	STA
$\delta(P,\ d^i)$	SP	$\delta(SP,\ pec^i)$	SP	$\delta(F,\ sf^i)$	TR
$\delta(P,\ pl^i)$	SP	$\delta(SP,\ rep^i)$	SP	$\delta(TA,\ per^i)$	SP
$\delta(P,\ pc^i)$	SP	$\delta(SP,\ mac^i)$	F	$\delta(STA,\ sa^i)$	TR
$\delta(P,\ ce^i)$	SP	$\delta(SP,\ r^i)$	P	$\delta(TR,\ exp^i)$	X

sustenta en una estructura de datos que maneja cada *jugada* y *estado* actual del AF. El algoritmo 1 describe la transición y secuencia de jugadas y 2 se muestran los subprocesos necesarios para hacer las transiciones mencionadas. Conviene observar que, sólo utilizando el AF, la ocurrencia de jugadas es correcta y acorde a la reglas del futbol, por ejemplo: una recuperación - un pase - una recepción - un pase - ... - un gol.

Algoritmo 1 Algoritmo general de transición en AF.

Entrada: Jugadas, Estados & ocurrencia de jugadas
1: Inicializa *estado actual* en q_0 de AF
2: **Mientras** *estado actual* $\neq F$ **Hacer**
3: Crear subconjunto $T \subset \Sigma$ con las jugadas posibles de transición
4: Elegir jugada $j \in T$ aleatoriamente de acuerdo a su ocurrencia
5: Determinar estado $s \in Q$ a transitar
6: Se crea la nueva transición con j hacia s
7: *estado actual* $= s$
8: **Fin Mientras**

El autómata descrito previamente da secuencias de jugadas para un jugador de FS. Sabemos que la ocurrencia de jugadas, si bien compartidas para *delanteros*, *mediocampistas* y *defensas*, es diferente en frecuencia de ocurrencia para cada rol de juego; el portero tiene un conjunto propio de jugadas. A continuación se muestra una cadena de jugadas para un delantero promedio:

$$pc^f\ r^f\ pl^f\ far^f\ far^f\ r^f\ ref^f\ far^f\ blce^f\ pc^f\ db^f\ r^f\ pc^f\ r^f\ pl^f\ r^f\ go^f$$ [1]

Para esta cadena de jugadas que finalizó en *gol* se siguieron las reglas de transición:

- $\delta(P,\ pc^i) = SP$
- $\delta(SP,\ r^i) = P$

[1] En este ejemplo se utilizó el superíndice f para denotar que se trata del delantero (*forward*, en inglés).

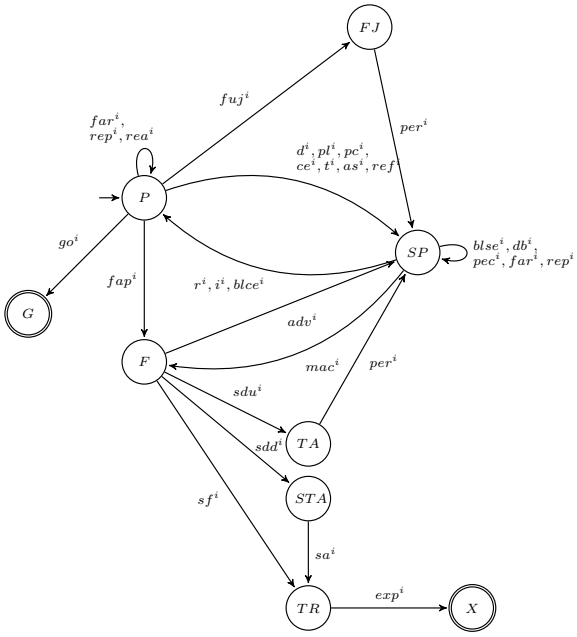


Fig. 1. Autómata Finito para un jugador de fútbol.

- $\delta(P,\ pl^i) = SP$
- $\delta(SP,\ far^i) = SP$
- $\delta(P,\ ref^i) = SP$
- $\delta(SP,\ blce^i) = P$
- $\delta(P,\ go^i) = G$
- ...

En contraste con una cadena de jugadas para un defensa promedio :

$pc^d\ r^d\ pc^d\ blce^d\ pc^d\ r^d\ pc^d\ r^d\ pc^d\ blce^d\ pc^d\ far^d\ blce^d\ pc^d\ r^d\ pc^d\ r^d\ pc^d\ blse^d\ r^d\ go^d$[2]

De acuerdo a los datos obtenidos, por ejemplo, un defensa realiza más jugadas de *pase corto*. Las secuencias de jugadas mostradas previamente se generaron a partir de los algoritmos, y corresponden a cadenas con pocas jugadas. Usualmente, en su mayoría, las cadenas obtenidas tienen más de 100 jugadas.

Debemos observar que las secuencias de jugadas ilustradas antes son sin considerar la frecuencia de ocurrencia real de cada jugada. Es necesario considerar las estadísticas de cada una de las jugadas, y combinar el análisis estadístico previo con las transicisones dadas por el AF.

[2] En este ejemplo se utilizó el superindice d para denotar que se trata del defensa(*defense*, en inglés).

Algoritmo 2 Algoritmo para elección de jugada aleatoria.

Entrada: Estado actual
Salida: Estado a transitar
1: Se crea conjunto T de jugadas posibles a transitar desde el *estado actual*
2: Se adapta la probabilidad de ocurrencia $\pi(j), j \in T$ a cada jugada
3: Definir variable de precisión flotante t (total)
4: **Para todo** $j \in T$ **Hacer**
5: $t = t + \pi(j)$
6: **Fin Para**
7: **Para todo** $j \in T$ **Hacer**
8: $\pi(j) = \pi(j)/t$
9: **Fin Para**
10: Ordenar T con respecto a $\pi(j)$
11: **Para todo** $j \in T$ **Hacer**
12: $\pi(j) = \pi(j) + \pi(j-1)$
13: **Fin Para**
14: Se genera un valor aleatorio r en el rango $[0, 1]$
15: **Para todo** $j \in T$ **Hacer**
16: **Si** $r \leq \pi(j)$ **Entonces**
17: *jugada actual* $= j$
18: **Fin Si**
19: **Fin Para**
20: Se determina el *estado a transitar* de acuerdo a la *jugada actual*

3.2. El fútbol como sistema concurrente

Para modelar un partido de FS en su dinámica colectiva se requiere un sistema concurrente. Cada hilo del sistema simula un jugador en el juego. La *región critica de concurrencia computacional,* se asocia a la posesión del balón. La región critica es compartida por los autómatas - hilos - jugadores, y el acceso a ella implica la posesión del balón, por un único jugador en cada jugada. El control de la región critica en el autómata es de tal forma que mientras un jugador se encuentre en el estado P(Posesión del balón) el resto de los jugadores están en el estado SP(Sin posesión del balón). La transición de un estado a otro es tal que si el jugador i en P hace un pase corto pc^i, y el jugador k hace una recuperación r^k, el autómata de cada jugador refleja la acción de él, y entre ambos modelan un movimiento concurrente, de pase corto, como se ilustra en la Figura 2. En *la región critica* se bloquea la transición si no se cumplan las condiciones para realizar un pase.

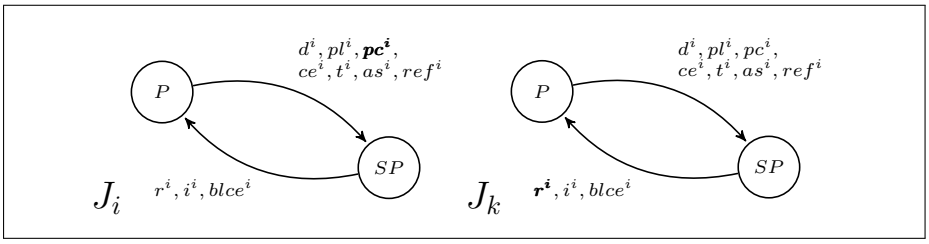


Fig. 2. El sistema concurrente de autómatas de 2 jugadores para un movimiento de pase corto.

La representación de concurrencia entre las jugadas, con o sin *posesión del balón*, destaca el modelado simultaneo de jugadas en un partido de futbol, y como cada jugada individual depende de las jugadas de los compañeros en la cancha. Similarmente, una *falta propia* (fap^i) la hace un jugador y otro *recibe falta* far^k, y se maneja en hilos concurrentes. Aún en juegos con una dinámica *pausada* como el béisbol, las jugadas suelen ocurrir concurrentes y en dependencia de las de otro(s) jugador(es): usualmente, un corredor avanza entre las bases 1 - 2 - 3 - *home* una vez que el bateador (*pitcher*) da un golpe válido a la bola, y éste de manera concurrente, avanza de *home* a 1, si bien, hay jugadas estrictamente secuenciales, e.g., el golpe a la bola del primer bateador y su avance a primera base, y más si el golpe fue *homerun*.

4. Conclusiones

La habilidad de un jugador en el FS, se caracterizan midiendo la *ocurrencia promedio de jugadas por minuto, OPM,* que dicho jugador realiza. La dinámica del FS se modela con autómatas concurrentes, un autómata por jugador, y la posesión del balón corresponde a la *región critica*. Con base en autómatas concurrentes y la OPM la simulación de un partido de FS con jugadores especificos, puede hacerse para equipos reales.

Referencias

1. Alvarado, M., Rendón, A.Y.: Nash equilibrium for collective strategic reasoning. Expert Syst. Appl. 39(15), 12014–12025 (2012), http://dx.doi.org/10.1016/j.eswa.2012.03.050
2. Alvarado, M., Rendón, A.Y., Cocho, G.: Simulation of baseball gaming by cooperation and non-cooperation strategies. Computación y Sistemas 18(4) (2014), http://cys.cic.ipn.mx/ojs/index.php/CyS/article/view/1987
3. McDougall, C.: Soccer. Best Sport Ever eBook Series, ABDO Publishing Company (2012), https://books.google.com.mx/books?id=2nrhG5ovUSQC
4. Rendón, A.Y., Rodríguez, R., Alvarado, M.: Analysis of strategies in american football using nash equilibrium. In: Agre, G., Hitzler, P., Krisnadhi, A.A., Kuznetsov, S.O. (eds.) Artificial Intelligence: Methodology, Systems, and Applications - 16th International Conference, AIMSA 2014, Varna, Bulgaria, September 11-13, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8722, pp. 286–294. Springer (2014), http://dx.doi.org/10.1007/978-3-319-10554-3_30

Un modelo de red bayesiana de la informalidad laboral en Veracruz orientado a una simulación social basada en agentes

Jean Christian Díaz-Preciado, Alejandro Guerra-Hernández, Nicandro Cruz-Ramírez

Universidad Veracruzana, Centro de Investigación en Inteligencia Artificial, Xalapa, Ver., Mexico

jean_christian12@msn.com, aguerra@uv.mx, ncruz@uv.mx

Resumen. La informalidad en el empleo en México es un fenómeno social de interés, ya que cerca de un 60 % de los trabajadores se desempeña en este sector. En este trabajo se propone un análisis de la informalidad en el empleo con la creación de un modelo de red bayesiana a partir de la base de datos generada de la Encuesta Nacional de Ocupación y Empleo, obtenida mediante el Instituto Nacional de Estadística y Geografía, con el propósito de utilizarla posteriormente en una simulación social basada en agentes y artefactos, en la cual los agentes obtengan algunas de sus propiedades de la base de datos y otras por inferencias a la red bayesiana generando datos artificiales. Se hace una comparación estadística de los datos generados en el sistema con los datos reales, lo cual se utilizará en un futuro para la validación de la simulación social bajo un paradigma lógico-estadístico.

Keywords: Base de datos, encuesta, redes bayesianas, agentes, validación estadística.

A Bayesian Network Model of Labour Informality in Veracruz Oriented to Agent-based Social Simulation

Abstract. In Mexico, informal employment is a social phenomenon of interest, given that about 60 % of the workers are in this situation. This paper presents an analysis of informal employment based a bayesian network model obtained from the data from the National Survey of Occupation and Employment, obtained by the National Institute of Statistics and Geography. The model is intended to be used in an agent-based social simulation, where agents get some properties directly from the data base and some others through bayesian inference, generating in this way artificial data. A statistical comparison of the data generated in the

system and the real data will be used in the future for validation of social simulation under a logical-statistical paradigm.

Keywords: Data base, survey, Bayesian network, agents, statistical validation.

1. Introducción

En México, el Instituto Nacional de Estadística y Geografía (INEGI) es el encargado de recabar información de personas, hogares y empresas, mediante la aplicación de encuestas y censos en periodos determinados, que tienen como objetivo principal proveer información para la generación de estadistícas que sirven como parámetro para la toma de decisiones en la implementación de políticas públicas. Estas encuestas y censos tienen diferentes temáticas, en este trabajo nos enfocamos en la Encuesta Nacional de Ocupación y Empleo (ENOE), que proporciona información sobre la ocupación de las personas, es decir, si tienen empleo o no; y en el caso de los empleados, que características tienen sus empleos. Los resultados publicados por el INEGI acerca de la informalidad laboral son presentados mediante indicadores y estadísticas con una cobertura geográfica nacional y estatal, desglosadas por sexo, edad y sector de la actividad [6].

El objetivo de este trabajo es construir un modelo que nos permita clasificar la situación laboral de una persona como formal o informal, dadas las prestaciones que proporciona su empleo y el contexto individual del trabajador. Se decidió que el modelo tomase la forma de una red bayesiana y se adoptó una aproximación de minería de datos basada en agentes para su construcción. Aunque esto no es mandatorio, nuestro interés por usar el modelo más adelante, en una simulación social basada en agentes, justifica la decisión. Hemos adoptado una aproximación de minería de datos basada en Agentes y Artefactos [8], muy similar a la usada en la herramienta JaCa-DDM [7]: Se provee una serie de artefactos basados en Weka [10], para almacenar datos, generar el modelo y evaluarlo. Los agentes usan estos artefactos en el proceso de aprendizaje. Puesto que el modelo y los datos son accesibles a los agentes, vía estos artefactos, los agentes pueden obtener algunas de sus propiedades de la base de datos y otras mediante inferencias bayesianas a partir del modelo. Los agentes reportan sus actividades generando una base de datos artificial, la cual es comparada estadísticamente con la base de datos real. Eventualmente nos gustaría modelar como afecta la implementación de políticas públicas la decisión laboral de los agentes, es decir, si optan por una situación formal o informal.

El artículo está organizado de la siguiente manera: En el capítulo 2 se muestran las características de la base de datos ENOE y la descripción de las variables utilizadas en este trabajo. En el capítulo 3 se hace la descripción del sistema que genera el modelo y los datos artificiales. Posteriormente, el capítulo 4 describe el diseño experimental y el capítulo 5 los resultados obtenidos del mismo. Finalmente se presentan las conclusiones y trabajo futuro, en los capítulos 6 y 7, respectivamente.

2. Base de datos ENOE

La base de datos ENOE está construída a partir de entrevistas realizadas en 120,000 viviendas repartidas en todo México, recabando información de alrededor de 800,000 personas. La información de la ENOE, puede estudiarse a nivel de vivienda, hogar y persona, la base de datos con la que se realizó el modelo fue obtenida de la tabla sociodemográfica (SDEMT110 [5]) del primer trimestre del año 2010. Los periodos posteriores podrán usarse para la validación de los datos artificiales generados.

El INEGI clasifica a la población en edad de trabajar legalmente, mayores de 15 años, en dos grandes grupos: Población Económicamente Activa (PEA), que es la que ejerce presión en el mercado laboral; y Población No Económicamente Activa (PNEA). Como podemos observar en la figura 1, la PEA se divide a su vez en Población Ocupada y Desocupada, según se tenga o no. Dentro de la población ocupada hay población sub ocupada refiriéndose a las personas que tienen un empleo pero continúan en busca de otro. La PNEA se divide en las personas disponibles, que aunque no están ocupadas ni buscando empleo al momento de la encuesta, bajo ciertas circunstancias podrían decidir incorporarse al mercado laboral; y las personas no disponibles, que son las que se encuentran bajo un contexto que les impide laborar.

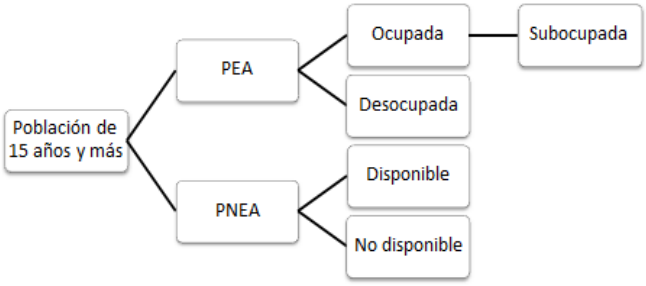


Fig. 1. Clasificación de la población [4].

Con la idea de atender la problemática regional, en este trabajo se considera la población sub ocupada que reside en el estado de Veracruz.

2.1. Descripción de variables

El cuadro 1 describe las variables consideradas para este trabajo, las cuales dividimos en generales y laborales. Las variables generales incluyen sexo, edad, escolaridad y si las personas estudian al momento de la encuesta; como se mencionó, nos enfocamos en las personas sub ocupadas, por ello se incluyen las variables que nos indican si se encuentran en busqueda de un nuevo empleo, y el motivo de la busqueda. Las variables laborales corresponden a algunas de

las características de los empleos, como es el nivel de ingreso, la duración de la jornada y la formalidad o informalidad del empleo. La variable de clasificación de empleo nos indica la infomalidad o formalidad del mismo, por ello es considerada como variable clase, ya que se quiere observar que tipo de empleo puede tener una persona dadas sus características.

Tabla 1. Variables seleccionadas para la creación del modelo [5].

Atributos generales			**Atributos laborales**		
Atributo	**Variable**	**Descripción**	**Atributo**	**Variable**	**Descripción**
Sexo	1	Hombre	Ingreso	1	Hasta 1 salario mínimo
	2	Mujer		2	De 1 a 2 salarios mínimos
Edad	1	14 a 24 años		3	De 2 a 3 salarios mínimos
	2	25 a 44 años		4	Des 3 a 5 salarios mínimos
	3	45 a 64 años		5	Mas de 5 salarios mínimos
	4	65 años y mas	Jornada	1	Ausente temporales
Escolaridad	1	Preescolar		2	Menos de 15 horas
	2	Primaria		3	De 15 a 34 horas
	3	Secundaria		4	De 35 a 48 horas
	4	Bachillerato		5	Mas de 48 horas
	5	Normal	Clasificación de empleos	1	Empleo informal
	6	Técnica		2	Empleo formal
	7	Profesional			
	8	Maestría			
	9	Doctorado			
Estudia actualmente	1	Sí			
	2	No			
Busca otro empleo	1	Sí			
	2	No			
Motivo de busqueda	1	Para tener otro empleo			
	2	Para cambiarse de empleo			
	3	No buscó			

Los nombres de las variables de la fuente original fueron cambiados para dar mayor claridad. Es importante especificar la relación que existe entre las variables generales y su representación como propiedades de los agentes que definiremos en nuestro sistema como trabajadores; así como la relación entre las variables laborales y su representación como propiedades de los artefactos que definiremos en nuestro sistema como empresas. El siguiente capítulo presenta la descripción detallada del sistema de Agentes y Artefactos propuesto.

3. Sistema multiagente para crear el modelo

Los agentes del sistema están basados en redes probabilísticas para el manejo de la incertidumbre. Se implementó una red bayesiana, la cual se define como un modelo probabilísitico representado mediante un grafo acíclico dirigido (GAD), en el cual los nodos representan las variables del fenómeno y las dependencias probabilistas que existan entre ellas se encuentran en la estructura del grafo. Asociada a cada nodo de la red hay una distribución de probabilidad condicional (TPC), dependiente de los nodos padre [9].

La razón de utilizar redes bayesianas es facilitar la interpretación del modelo mediante el GAD y que nos permite observar la probabilidad que tiene una persona de tener un empleo formal o informal, y de las características del empleo, según su edad, sexo y escolaridad. Esto se realiza por medio de inferencias al modelo enviando como evidencia las variables generales [3].

Dado que el sistema está basado en Agentes y Artefactos, es deseable que la red bayesiana sea accesible a los agentes, ya sea como parte de ellos o como parte del algún artefacto. Para construir el modelo, hemos extendido JaCa-DMM [7], definiendo una nueva estrategia de aprendizaje basada en redes bayesianas y agregando un artefacto que encapsula SamIam [1], para realizar las inferencias. La figura 2 muestra el diagrama general del sistema, que tiene como entrada una base de datos y una red bayesiana que puede ser ingresada manualmente o generada por el sistema a partir de los datos. A continuación se describen a detalle las tareas realizadas por los artefactos y agentes en el sistema, en la figura 3 podemos observar el diagrama de los procesos realizados por estos.

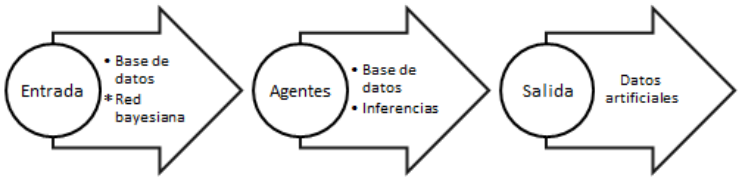


Fig. 2. Diagrama general del sistema multiagente.

3.1. Artefactos

El sistema cuenta con 3 artefactos que realizan tareas correspondientes a la minería de datos, los cuales se describen a continuación:

InstancesBase Este artefacto se adopta directamente de la herramienta JaCa-DDM, que a su vez la adopta de Weka. Se trata de un repositorio de ejemplos de entrenamiento que puede cargar archivos ARFF para su uso en clasificadores, evaluadores y demás herramientas Weka. Al tomar la forma de artefacto, los ejemplos y sus atributos son accesibles a los agentes del sistema y a otros artefactos.

BayesNet Este artefacto encapsula los métodos de construcción de redes bayesianas implementados en Weka. Su tarea principal es construir modelos a partir de datos y ponerlos a disposición del artefacto basado en SamIam. Dependiendo de las entradas del sistema realiza las siguientes tareas:

- Leer una red bayesiana generada previamente en formato XMLBIF

- Generar una red bayesiana a partir de los datos del artefacto InstancesBase y crear un modelo con los parámetros descritos en el capítulo 4 (Diseño experimental). Estos parámetros son fijos. Una vez generado el modelo, éste se guarda en formato XMLBIF.

SamIam Es el artefacto que los agentes usan para hacer inferencias basadas en una red bayesiana, la generada con el artefacto BayesNet. Por medio de las herramientas de SamIam, envía los parámetros para tener acceso a las tablas TPC; Recibe las evidencias de los agentes trabajadores para hacer la inferencia de las variables no conocidas, poniendo a disposición de los agentes las TPC obtenidas.

3.2. Agentes

El sistema cuenta con tres clases de agentes, dos se utilizan para crear agentes que controlan los experimentos y una para crear agentes que representan trabajadores. A continuación se describen las tareas que realiza cada agente:

Agente learning El sistema inicia su ejecución con este agente, el cual tiene como meta realizar las tareas correspondientes a la minería de datos por medio de los artefactos, para que los demás agentes tengan acceso a la base de datos y al modelo. Para poder concretar su meta, crea los artefactos y establece las ligas necesarias entre ellos. Su plan sigue esta secuencia:

- Leer base de datos (IntancesBase),
- Generar modelo (BayesNet),
- Preparar modelo para inferencias (SamIam).

El agente está diseñado para iniciar sus acciones ya sea recibiendo la base de datos y crear la red bayesiana; O recibir un modelo generado previamente. Al concluir sus tareas, envía un mensaje al agente control para que este comience sus actividades.

Agente control La meta de este agente es crear a los agentes que representan a los trabajadores a partir de los datos almacenados en el artefacto InstancesBase. Su única creencia es el número de personas que debe crear. Al momento de crear un nuevo agente le proporciona un nombre, que corresponde al número de caso de la base de datos, para que el nuevo agente obtenga sus propiedades generales de éste. También es el encargado de controlar el acceso al artefacto SamIam al momento de las inferencias de los agentes que representan los trabajadores.

Agente persona Esta clase de agente se usa para representar trabajadores, por lo que su meta principal es instanciar sus propiedades generales y laborales. Para las propiedades generales, recupera sus datos almacenados en el artefacto

InstancesBase a partir de su nombre. Las propiedades laborales son inferidas en el artefacto SamIAm, con base en la evidencia que proveen las propiedades generales del agente y el modelo almacenado en el artefacto BayesNet. Puesto que las TPC generadas tienen 2 o más variables, se ejecuta una acción interna que representa la ruleta propuesta por DeJong [2] y según su probabilidad condicional, se determina la propiedad laboral del agente. Una vez obtenidas todas sus propiedades, estas son almacenadas en un archivo de texto.

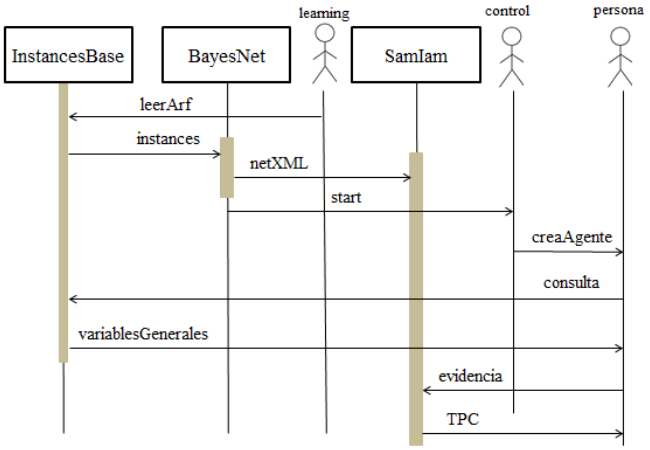


Fig. 3. Diagrama de procesos del sistema multiagente.

4. Diseño experimental

El sistema multi-agente propuesto, proporciona como salida una base de datos artificial y, si se desea, la red bayesiana generada de la base de datos. Como se mencionó, se determinó construir este modelo a partir de las personas sub ocupadas con residencia en el estado de Veracruz, las cuales en la ENOE del primer trimestre del 2010 constituyen un universo de 595 casos.

El cuadro 2, muestra los parámetros implementados para la generación de la red bayesiana, los cuales se determinaron en base a la observación de las redes generadas en experimentos utilizando Weka. En estos experimentos se observó que los parámetros que modificaban significativamente al modelo fueron el iniciarlo con un modelo Naive y la implementación de la manta de Markov, ya que sin ella, dos variables consideradas como relevantes, la edad y si el trabajador está estudiando, quedaban fuera de la red. Se llevó a cabo una validación cruzada del model con diez pliegues.

Dado que las variables generales son obtenidas directamente de la ENOE, la validación de los datos artificiales se realiza comparando la distribución de las variables laborales. Posteriormente se generó una red bayesiana con la base de

Tabla 2. Parámetros para generar el modelo en Weka.

Estimador	Simple Estimador	A 0.5
Algoritmo de Busqueda	HillClimber	
Parámetros del Algoritmo de busqueda	initNaiveBayes	False
	MarkovBlanket Classifier	True
	mxNrOfParents	10,000
	scoreTYPE	MDL
	useArcReversal	True

datos artificial la cual se comparó con la base de datos generada por el sistema para complementar la validación.

5. Resultados

El modelo generado por el sistema a partir de los datos reales, es muy similar al generado por Weka con los mismos datos. Con los datos artificiales se generó un modelo en Weka usando los mismo parámetros. El modelo obtenido con los datos de la ENOE se muestra en la figura 4(a), y el modelo generado por los datos artificiales en la figura 4(b). Los resultados estadísticos calculados por Weka se pueden observar en el cuadro 3.

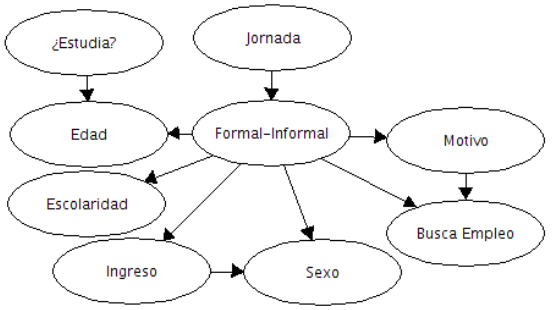
Tabla 3. Comparación de estadísticas de la generación del modelo.

	Datos reales	**Datos Artificiales**
Porcentaje de clasificados correctamente	74.11 %	63.80 %
Desviación Estándar	5.46	6.13
Área bajo la curva ROC	0.8180	0.6383

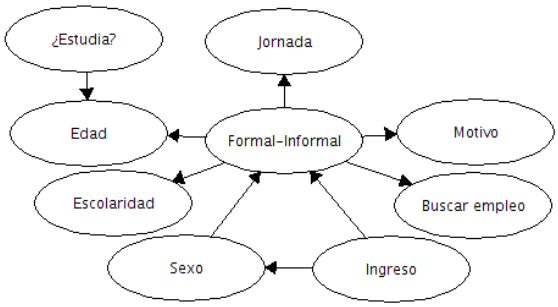
Se generaron las gráficas de curva ROC, con umbral de 0.5, para cada variable de la clase, con ambas bases de datos. Las gráficas ROC para la variable de informalidad en el empleo se muestran en las Figura 5(a) para los datos reales y en la Figura 5(b) para los artificiales, y para la variable de formalidad en el empleo en las Figuras 5(c) y 5(d).

También se realizó una comparación de las distribuciones de las variables que se obtuvieron en las inferencias al modelo. Dado que se generó el mismo número de agentes que de casos de la base de datos, la distribución de las variables generales son las mismas. En el Cuadro 4 se muestran las estadísticas de las variables laborales.

La Figura 6 muestra la comparación gráfica de la distribución de las variables. Los datos generados por medio de las inferencias y la aplicación de la ruleta, nos proporcionan resultados muy parecidos a los reales.

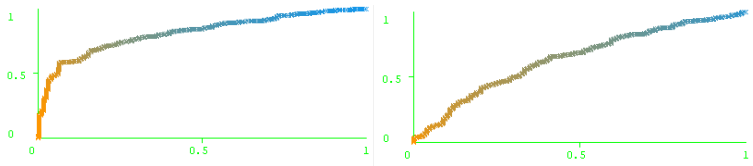


(a)

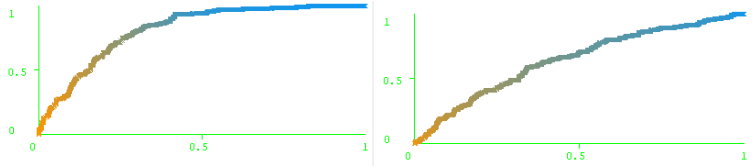


(b)

Fig. 4. Modelos de red bayesiana obtenidos: (a) Datos reales y (b) Datos artificiales.



(a) Datos reales (b) Datos artificiales



(c) Datos reales (d) Datos artificiales

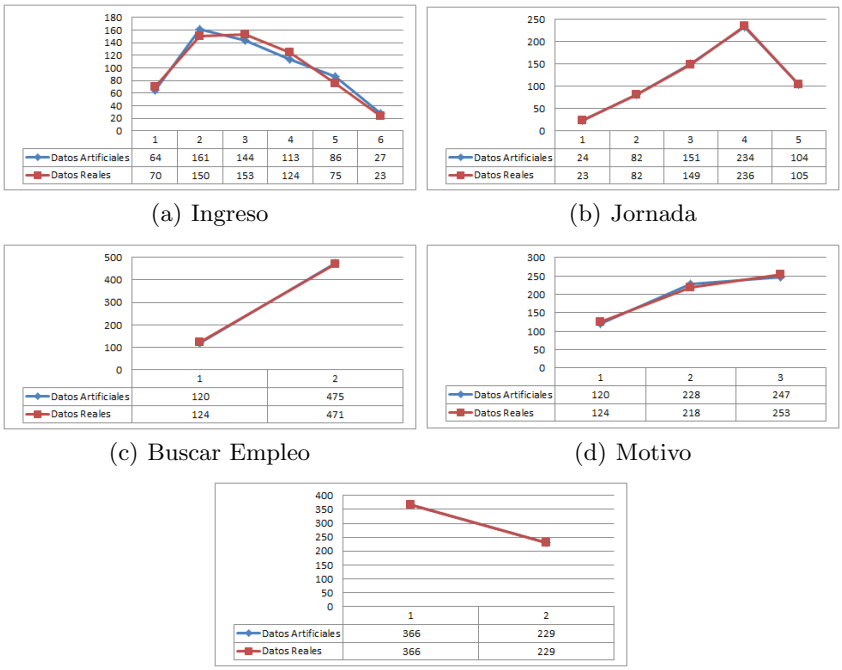
Fig. 5. Curvas ROC de los modelos generados.

5.1. Informalidad en el Estado de Veracruz

Como se mencionó, uno de los principales objetivos del trabajo del INEGI es dotar de estadísticas e información a los órganos encargados de la generación de

Tabla 4. Comparación de estadísticas.

Variable	Media		EE Media		DesvEst		Varianza		Asimetria		Curtosis	
	Real	Artificial	Real	Artificial	Real	Artificial	Real	Artificial	Real	Artificial	Real	Artificial
Ingreso	3.0891	3.1294	0.0547	0.0560	1.3335	1.3652	1.7782	1.8637	0.25	0.28	-0.71	-0.80
Jornada	3.5345	3.5244	0.0432	0.0434	1.0541	1.0576	1.1112	1.1185	-0.49	-0.48	-0.38	-0.39
Buscar	1.7916	1.7983	0.0167	0.165	0.4065	0.4616	0.1652	0.1613	-1.44	-1.49	0.07	0.22
Motivo	2.2168	3.2134	0.0314	0.0310	0.7665	0.7564	0.5876	0.5722	-0.39	-0.38	-1.21	-1.17
Formal/Informal	1.3849	1.3849	0.0200	0.0200	0.4870	0.4870	0.2371	0.2371	0.47	-1.78	-1.78	-1.78



(a) Ingreso (b) Jornada

(c) Buscar Empleo (d) Motivo

(e) Informalidad

Fig. 6. Comparación de distribución de variables.

políticas públicas. Estas estadísticas se presentan como gráficas con 2 variables a observar, por ejemplo la cantidad de personas por sexo que tienen un empleo formal o informal.

En este trabajo se realizó el análisis de los datos por medio de consultas al modelo, observando la propagación de las probabilidades dados ciertos valores de las variables que se observaron. Es importante mencionar que para este experimento, no se consideraron todas las variables que provee la base de datos ENOE y solo se trabajó con las personas subocupadas del Estado de Veracruz. A continuación se presentan tres consultas generadas:

- La variable clasificiación de empleo es dependiente del número de horas que trabaja una persona, se observó que solo en el rango de 35 a 48 horas laborales, hay más personas con empleos formales que informales, figura 7.

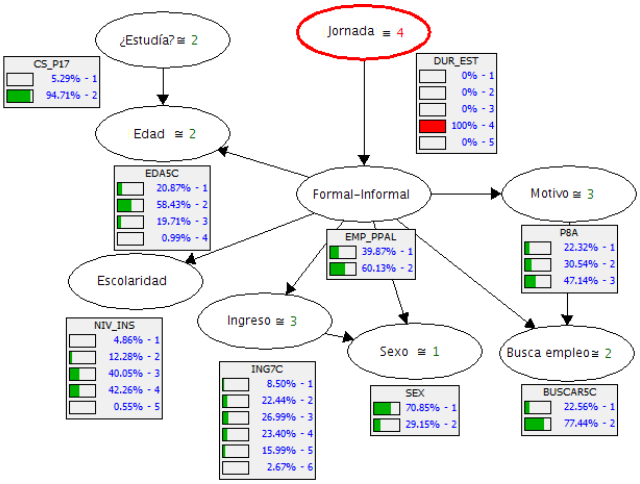


Fig. 7. Propagación por Jornada Laboral [1].

- Para los empleos informales se observó que las personas cuentan con un menor nivel escolar, los sueldos son menores y hay más personas en busca de un nuevo empleo para dejar el actual, Figura 8.

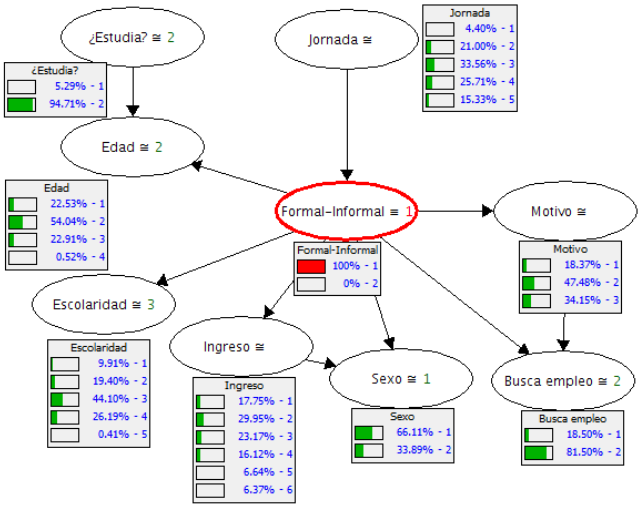


Fig. 8. Propagación por Informalidad en el Empleo [1].

- Para los empleos formales se observó que las personas tienen un mayor nivel escolar, los sueldos se sitúan en rangos más altos y es menor el número de

persona en busca de otro empleo, sin embargo las que están en busca de otro empleo es en su mayoría por que desean dejar su empleo actual, figura 9.

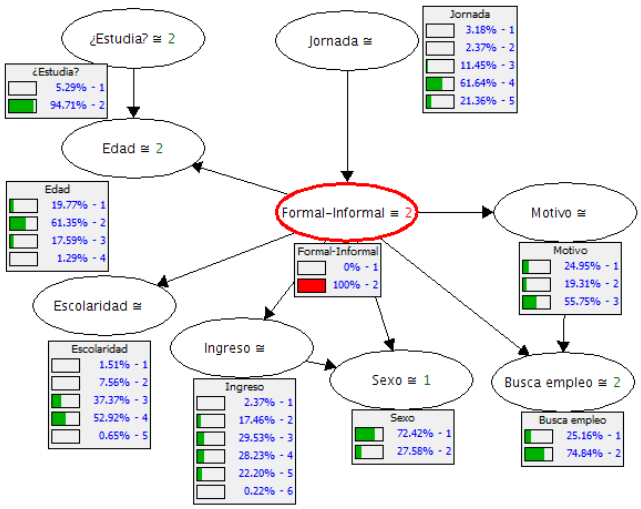


Fig. 9. Propagación por Formalidad en el Empleo [1].

6. Conclusiones

Podemos observar como la red generada con los datos artificiales, es muy similar a la generada con los datos reales, considerando que el valor de las variables que adquieren los agentes como propiedades laborales, está determinado por una ruleta, se da oportunidad de adquirir valores que no tengan la probabilidad más grande. Las diferencias en los GAD de los modelos se describen en el cuadro 5.

Tabla 5. Diferencia entre modelos.

Datos reales	Datos artificiales
El nodo Jornada es independiente	El nodo Jornada es dependiente del nodo de clasificación del empleo
El nodo de clasificación del empleo es dependiente del nodo Jornada	El nodo de clasificación del empleo es dependiente de los nodos Ingreso y Sexo

A pesar que los datos con los que se genera el modelo son solo 595, y el número de nodos de la red son 9, la distribución de las variables que se obtienen

por medio de inferencias tienen una gran similitud con las reales. Consideramos que el porcentaje de casos clasificados correctamente con el modelo generado es bueno teniendo 74.11 % con una desviación estándar de 5.46, y a pesar de que las distribuciones de las variables de los datos artificiales son similares, el porcentaje que se obtuvo es de 63.80 %, con una desviación estándar de 6.13, evidentemente el porcentaje disminuye más de un 10 %, sin embargo la desviación estándar no es muy alta, por lo que se espera tener mejoras experimentando con bases de datos que contemplen un número mayor de casos.

Se generó el área bajo la curva ROC, para visualizar como los datos artificiales tienden a disminuir la generalidad del modelo, y con esto a cometer errores en la clasificación. Estos resultados nos proporcionan un panorama del número de agentes que debe tener nuestra simulación así como la complejidad de la red bayesiana que se implementará, para mantener la consistencia que nos proporciona el modelo en nuestros datos artificiales. Las comparaciones estadísticas que observamos en el cuadro 4 nos muestran las similitudes entre las bases de datos. Observamos que el atributo correspondiente al ingreso es el que tiene mayor diferencia en la distribución de los datos, esto se debe al número de variables que tiene y la distribución de las mismas, que observamos en la desviación estándar y la varianza, como complemento se calculó la curtosis que es de -0.71 y el valor de asimetria de 0.25; el atributo jornada también tiene un número mayor de variables, pero no se presentan cambios significativos en los datos ya que tiene una desviación estándar menor, y tiene una curtosis de -0.39 con un valor de asimetría de -0.48. Los atributos que solo tienen dos o tres variables presentaron las menores diferencias en la comparación, teniendo una distribución igual en la variable correspondiente a la formalidad o informalidad del empleo. El error cuádratico medio (EMC) calculado es de 0.81, con lo cual se determinó que los resultados son aceptables para el caso de estudio.

La implementación de la minería de datos en un entorno de agentes y artefactos, es una herramienta útil que nos facilita observar el flujo de trabajo que se realiza. Los artefactos nos proporcionan la distribución de las tareas que se requieran efectuar, ya sea desde la lectura de la base de datos para la generación del modelo o partiendo de un modelo ya generado. Los agentes se utilizaron para definir el orden en que se deben efectuar los procesos. Los resultados de la predicción mediante inferencias a redes bayesianas nos proporcionan datos aproximados a los datos con los que se genera el modelo. La determinación de analizar una base de datos obtenida de encuestas a personas con redes bayesianas, nos proporciona información sobre la causalidad del fenómeno a observar.

7. Trabajo futuro

En un futuro se prentede utilizar las inferencias que realizan los agentes al modelo en una simulación social, en donde el valor de los atributos que obtengan influyan en su toma de decisiones. Dicho esto se probará las inferencias con bases de datos con más casos, y con periodos de tiempo definidos, por ejemplo, observar el comportamiento de las variables en un periodo de un año equivalente a cuatro

encuestas de la ENOE. Como se mencionó la ENOE puede observarse a nivel hogar, por lo que se pretende hacer experimentos a este nivel, representando los hogares por medio de artefactos, con el fin que los agentes que pertenezcan a un mismo hogar tengan acceso a la información de los integrantes de su familia. Se pretende implementar por medio de artefactos, una abstracción de políticas públicas, es decir, de qué forma afectan a un hogar o individuo, y agregar comportamientos de los agentes con respecto a los cambios con los que se enfrentará.

Agradecimientos. El primer autor cuenta con el apoyo de la beca CONACyT número 633473.

Referencias

1. Darwiche, A.: Modeling and Reasoning with Bayesian Networks. Cambridge University Pres, New York (2009)
2. De Jong, K.A.: Analysis of the behavior of a class of genetic adaptive systems. Tech. Rep. 185, The University of Michigan (1975)
3. Glymour, C.: Discovering Causal Structure. Academic Press, Orlando, Florida (1987)
4. INEGI: Conociendo la base de datos de la ENOE. Datos ajustados a proyecciones de población 2010. INEGI (2010)
5. INEGI: ENOE. Descripción de Archivos. INEGI (2010)
6. INEGI: México: Nuevas estadísticas de informalidad laboral. INEGI (2013)
7. Limón, X., Guerra-Hernández, A., Cruz-Ramírez, N., Grimaldo, F.: An agents and artifacts approach to distributed data mining. In: Advances in Soft Computing and Its Applications, pp. 338–349. Springer (2013)
8. Omicini, A., Ricci, A., Viroli, M.: Artifacts in the A&A meta-model for multi-agent systems. Autonomous Agents and Multi-Agent Systems 17(3), 432–456 (2008)
9. Pearl, J.: Probabilistic Reasoning in Intelligence Systems. Morgan Kauffman, San Mateo, CA (1988)
10. Witten, I.H., Frank, E.: Data mining, Practical Machine Learning Tools and Techniques. Morgan Kauffman, San Francisco, CA (2011)

Sintonizador fuera de línea de un controlador PID discreto usando un algoritmo genético multiobjetivo

David Bedolla-Martínez[1], Esther Lugo González[2],
Felipe Trujillo-Romero[1], F. Hugo Ramírez Leyva[3]

[1] División de Estudios de Posgrado, Universidad Tecnológica de la Mixteca,
Huajuapan de León, Oaxaca,
México

[2] CONACYT-Universidad Tecnológica de la Mixteca, Instituto de Electrónica y Mecatrónica, Universidad Tecnológica de la Mixteca,
Huajuapan de León, Oaxaca,
México

[3] Instituto de Electrónica y Mecatrónica, Universidad Tecnológica de la Mixteca, Huajuapan de León, Oaxaca,
México

davidbedollamartinez@outlook.es, {elugog, ftrujillo, hugo}@mixteco.utm.mx

Resumen. Este trabajo presenta los resultados obtenidos, al usar un algoritmo genético multiobjetivo, para la sintonización de las ganancias de un controlador PID para el control de posición. Para evaluar este enfoque, la planta a controlar es un Robot Manipulador de 2 Grados de Libertad (GDL), que cuenta con las siguientes características: El modelo dinámico es No-Lineal, es un sistema de múltiples entradas y múltiples salidas (MIMO) y se tiene un acoplamiento en las dinámicas de cada articulación. Como resultado se obtuvo un conjunto de ganancias que estabilizan al sistema con un sobreimpulso y tiempo de establecimiento relativamente bajos.

Palabras clave: PID, algoritmo genético multiobjetivo, robot manipulador, sintonizador fuera de línea.

Offline tuner of a discrete PID controller using a multi-objective genetic algorithm

Abstract. This paper presents the results obtained when using a multi-objective genetic algorithm for tuning the gains of a PID controller for position control. To evaluate this approach, the plant to control is a robot manipulator with 2 degrees of freedom (DOF), which has the following characteristics: The dynamic model is not linear, it is a system of multiple input and multiple output (MIMO) and it has a coupling on the dynamics of each joint. As a result a set of gains that stabilize the system with an overshoot and settling time relatively low was obtained.

Keywords: PID, multi-objective genetic algorithm, robot manipulator, offline tuner.

1. Introducción

Sintonizar controladores de tipo Proporcional + Integral + Derivativo (PID), no es trivial, y a pesar de que existen diversos estudios para sintonizar estos controladores, parece no haberse resuelto el problema para tener una técnica que pueda ser implementada en los diversos sistemas, e.g. Los sintonizadores clásicos están limitados a sistemas de una entrada y una salida (SISO), de naturaleza lineal y que son estables en lazo abierto o los métodos inteligentes como las redes neuronales están limitados a los recursos computacionales [1]. El controlador PID no es el más adecuado para ciertos sistemas no lineales. Sin embargo, es el controlador preferido por la industria por la facilidad de implementarlos [2].

Para un controlador no es suficiente con mantener la respuesta deseada, también se pide que éste converja en un tiempo finito, otra manera de ver este acontecimiento, es llevar el error a cero en un lapso muy corto de tiempo sin tener grandes sobreimpulsos en la respuesta.

Estos requerimientos se modifican con un controlador PID y se varían a partir de sus ganancias, i.e., a partir de una configuración de ganancias, se tienen diferentes tiempos de establecimiento, sobreimpulso máximo de la planta, etc. Por tanto el objetivo es la búsqueda de ganancias que produzcan los requerimientos de diseño de tiempo y sobreimpulso mínimo.

Aunque los controladores PID son ampliamente usados en los procesos industriales, su efectividad es frecuentemente limitada debido a una sintonización pobre. La sintonización manual de las ganancias de un controlador es una tarea que consume demasiado tiempo [3]. Algunas investigaciones sobre la sintonización de controladores PID para sistemas MIMO se describen a continuación:

W.D. Chang [4] propone una modificación de la fórmula de recombinación de un algoritmo genético y este método se usa para determinar las ganancias de los controladores PID en proceso multivariables, i.e. sistemas MIMO. Donde se asegura que el algoritmo genético es uno de los métodos convincentes de búsquedas óptimas para la solución de problemas de control.

En [5] se presenta una comparación del desempeño de los algoritmos evolutivos para la sintonización de los controladores PIs y PIDs multivariables, tales algoritmos son: algoritmos genéticos de código real (RGA), optimización por cúmulo de partículas modificado (MPSO), la adaptación de la matriz de covarianza con estrategia de evolución (CMAES) y evolución diferencial (DE). A través de las simulaciones realizadas revelan que los cuatro algoritmos considerados son adecuados para la sintonización del controlador PID en modo fuera de línea. Sin embargo solo los algoritmos CMAES y MPSO son adecuados para la sintonización en línea del controlador PID.

En [6] se implementó un controlador PID para manipular la velocidad de un motor de CD, donde el controlador fue sintonizado con un algoritmo genético e implementado en una FPGA. Sung-Kwun en [7] propone controladores difusos para estabilizar el sistema Viga-Bola, se compara con controladores PDs. La sintonización de los

parámetros de estos controladores se realizó mediante un algoritmo genético basado en la competencia leal jerárquica (HFCGA, por sus siglas en inglés, hierarchical fair competition-based genetic algorithm). En [8] se sintoniza un controlador PID mediante un algoritmo genético multiobjetivo para un robot manipulador.

La problemática es la sintonización fuera de línea de 2 controladores PID para un robot de 2 GDL, el cual es un sistema no lineal tipo MIMO. Esta sintonización se realizará a partir de proponer las ganancias de los controladores, observando la respuesta del sistema con los controladores, evaluando el sobreimpulso máximo y tiempo de establecimiento repitiendo el proceso. Para automatizar este proceso se hará uso de un algoritmo genético multiobjetivo debido a que se requiere buscar el mínimo global de 4 variables, que son dos sobre impulsos máximos y dos tiempos de establecimiento.

2. Sintonización fuera de línea

a. Robot Manipulador

En la Fig. 1 se observa un manipulador Robótico, el cual consta de eslabones y articulaciones. Los eslabones sirven de soporte a las articulaciones que es en donde se genera el movimiento. La posición final del efector del robot está en función de la posición angular que adquiere cada articulación (coordenadas articulares). La posición final del robot **X** es un vector con tres componentes que se muestran en la ecuación (1):

$$\boldsymbol{X}^T = [x_1 \quad x_2 \quad x_3]^T. \tag{1}$$

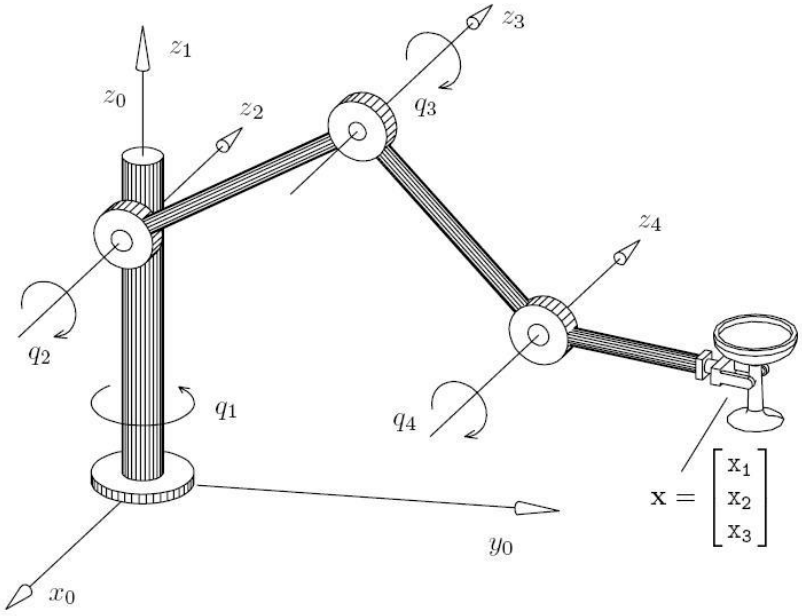


Fig. 1. Robot manipulador [9].

En forma general un robot se compone de n articulaciones, por lo cual el vector de posición $\boldsymbol{q}$ está dado por la ecuación (2), donde cada término corresponde a la coordenada de cada articulación:

$$\boldsymbol{q^T} = [q_1 \quad q_2 \quad \cdots \quad q_n]^T. \tag{2}$$

Al generarse el movimiento del robot se tiene n velocidades que corresponden a la derivada de la posición y se muestran en la ecuación (3), donde $\dot{\boldsymbol{q}}$ corresponde a dicho parámetro:

$$\dot{\boldsymbol{q}}^{\boldsymbol{T}} = [\dot{q}_1 \quad \dot{q}_2 \quad \cdots \quad \dot{q}_n]^T. \tag{3}$$

Normalmente en cada articulación se tiene un motor que genera el par mecánico para mover la estructura mecánica. Para simplificar al sistema no se toma en cuenta la dinámica de éste y únicamente se usa como entrada el vector de par $\boldsymbol{\tau}$ de cada articulación, que está dado en la ecuación (4):

$$\boldsymbol{\tau^T} = [\tau_1 \quad \tau_2 \quad \cdots \quad \tau_n]^T. \tag{4}$$

La posición $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ velocidad y $\boldsymbol{\tau}$ el par en cada articulación forma un vector. Mientras $\boldsymbol{X}$ es la posición del efector final.

Con base en la cinemática se determina el espacio de trabajo utilizando los parámetros Denavit-Hartenberg. En el modelado dinámico se obtienen las ecuaciones diferenciales que gobiernan el funcionamiento del sistema.

El modelado dinámico se obtiene a partir de las ecuaciones de Euler-Lagrange y el procedimiento para obtenerlo es el siguiente:

1. Obtener la cinemática directa

$$\boldsymbol{X} = f(q_1, q_2, \ldots q_n). \tag{5}$$

2. Modelo de energía

- Calculo de la energía cinética K
- Calculo de la energía potencial U

3. Cálculo del Lagrangiano

$$L = K - U. \tag{6}$$

4. Desarrollo de las ecuaciones de Euler-Lagrange

$$\frac{d}{dt}\left(\frac{\partial L(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \dot{\boldsymbol{q}}}\right) - \frac{\partial L(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \boldsymbol{q}} = \boldsymbol{\tau}. \tag{7}$$

La ecuación (7) representa a todas las ecuaciones diferenciales del sistema. Si el robot tiene 2 articulaciones esto implica que se tienen 2 ecuaciones diferenciales.

Si se utilizan directamente las ecuaciones obtenidas con Euler-Lagrange resultan ecuaciones diferenciales acopladas con complicada manipulación. Un enfoque

alternativo es usar una representación general del sistema como se muestra en la ecuación (8) la cual permite desacoplar directamente las ecuaciones diferenciales [10].

Un robot Manipulador de 2 GDL se muestra en la Fig. 2. Está compuesto de 2 articulaciones, 2 eslabones y dos entradas de control. Los parámetros del sistema son: m_1 la masa del eslabón 1, l_1 la longitud del eslabón 1, I_1 la inercia del eslabón 1, l_{c1} el centro de masa del eslabón 1, q_1 la posición articular del eslabón 1, m_2 la masa del eslabón 2, l_2 la longitud del eslabón 2, I_2 la inercia del eslabón 2, l_{c2} el centro de masa del eslabón 2, q_2 la posición articular del eslabón 2.

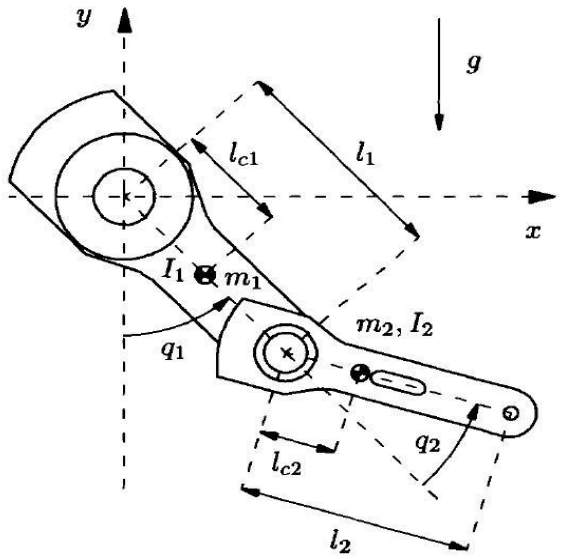


Fig. 2. Robot manipulador de 2 GDL.

El modelo dinámico se obtuvo de [11] y se muestra en la ecuación (8). Sus términos son: La matriz de Inercia $\boldsymbol{M}(\boldsymbol{q})$ y matriz de Coriolis $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ son de 2x2, el vector de par gravitacional $\boldsymbol{g}(\boldsymbol{q})$, fricción $\boldsymbol{f_f}(\dot{\boldsymbol{q}})$ son vectores de 2x1.

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{f_f}(\dot{\boldsymbol{q}}) = \boldsymbol{\tau},$$

$$\mathbf{M}(\boldsymbol{q}) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix},$$

$$\boldsymbol{C}(\boldsymbol{q}) = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}, \tag{8}$$

$$\boldsymbol{g}(\boldsymbol{q}) = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix},$$

$$\boldsymbol{f_f}(\dot{\boldsymbol{q}}) = \begin{bmatrix} f_{f1} \\ f_{f2} \end{bmatrix}.$$

Los parámetros del modelo son función de las coordenadas articulares, velocidades y posiciones articulares y los parámetros físicos del sistema están en las ecuaciones de la (9) a la (20), los cuales son función de los vectores de posición y velocidad articular. Donde la función $sat(\tau, f_e)$ que depende del par de entrada y del coeficiente de fricción estática, representa el signo y magnitud del coeficiente de fricción estática.

Los parámetros que se muestran en la Tabla 1 pertenecen al robot manipulador de 2 GDL:

$$M_{11} = m_1 l_{c1}^2 + m_2[l_1^2 + l_{c2}^2 + 2 l_1 l_{c2} cos(q_2)] + I_1 + I_2, \tag{9}$$

$$M_{12} = m_2[l_{c2}^2 + l_1 l_{c2} cos(q_2)] + I_2, \tag{10}$$

$$M_{21} = m_2[l_{c2}^2 + l_1 l_{c2} cos(q_2)] + I_2, \tag{11}$$

$$M_{22} = m_2 l_{c2}^2 + I_2, \tag{12}$$

$$\mathrm{C}_{11} = -m_2 l_1 l_{c2}\, \mathrm{sen}(q_2)\, \dot{q}_2, \tag{13}$$

$$\mathrm{C}_{12} = -m_2 l_1 l_{c2} sen(q_2)[\dot{q}_1 + \dot{q}_2], \tag{14}$$

$$\mathrm{C}_{22} = 0, \tag{15}$$

$$\mathrm{g}_1 = [m_1 l_{c1} + m_2 l_1] g sen(q_1) + m_2 l_{c2} g sen(q_1 + q_2), \tag{16}$$

$$\mathrm{g}_2 = m_2 l_{c2} g sen(q_1 + q_2), \tag{17}$$

$$f_{f1} = b_1 \dot{q}_1 + f_{c1} signo(\dot{q}_1) + [1 - |signo(\dot{q}_1)|] sat(\tau_1; f_{e1}), \tag{18}$$

$$f_{f2} = b_2 \dot{q}_2 + f_{c2} signo(\dot{q}_2) + [1 - |signo(\dot{q}_2)|] sat(\tau_2; f_{e2}), \tag{19}$$

$$sat(\tau, f_e) = \begin{cases} f_e & si\ \tau > f_e \\ \tau & si - f_e \le \tau \le f_e. \\ -f_e & si\ \tau < -f_e \end{cases} \tag{20}$$

Tabla 1. Parámetros del robot de 2 GDL.

Parámetro	Valor
l_1	0.45 m
l_{c1}	0.091 m
l_2	0.45 m
l_{c2}	0.048 m
m_1	23.902 kg
m_2	3.88 kg
I_1	1.266 Nm seg^2/rad
I_2	0.093 Nm seg^2/rad
b_1	2.288 Nm seg/rad
b_2	0.175 Nm seg/rad
τ_1	150 Nm
τ_2	15 Nm
g	9.81 $m/_{s^2}$

b. Algoritmo Genético Multiobjetivo

Para sintonizar las ganancias de dos controladores PID se usó un algoritmo genético multiobjetivo, el cual busca disminuir el sobreimpulso máximo y el tiempo de establecimiento con un criterio del 2% de las 2 articulaciones del robot manipulador.

El algoritmo genético multiobjetivo se codificó usando toolbox de Matlab ®. Algunos criterios para la implementación del algoritmo genético son:

- El tipo de población es double,
- El tamaño de la población es de 10,
- El tipo de selección es con torneos con un tamaño de 2,
- La probabilidad de cruce es de 0.8,
- La probabilidad de mutación es de 0.001,
- El número de generaciones es de 100,
- El rango de búsqueda para la ganancia proporcional es de [0 300],
- El rango para la ganancia integral es de [0 600],
- El rango para la ganancia derivativa es de [0 50].

Las características anteriores dieron un tiempo de simulación de 2 minutos. La primera vez que se ejecutó el algoritmo genético tomó cerca de una hora debido a que las características eran diferentes como se muestra a continuación:

- El tamaño de la población era de 100,
- El número de generaciones era de 600,
- El rango de búsqueda para la ganancia proporcional era de [0 600],
- El rango para la ganancia integral era de [0 600],
- El rango para la ganancia derivativa era de [0 600].

Esto permitió reducir el rango de búsqueda de ganancias, el tamaño de la población y el número de generaciones con el fin de reducir el tiempo de simulación.

c. Metodología

Para cada individuo se proponen 6 ganancias 3 para cada controlador PID, se simula la planta con los controladores en lazo cerrado, posteriormente se evalúa el sobreimpulso máximo y el tiempo de establecimiento. En la Fig. 3 se muestra el diagrama del algoritmo genético.

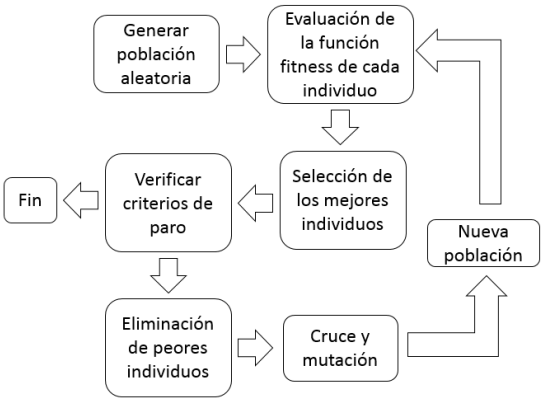


Fig. 3. Diagrama del Algoritmo genético.

La función fitness define 4 valores que son el sobreimpulso y el tiempo de establecimiento para ambas articulaciones, en la Fig. 4 se puede ver el diagrama de la función fitness. Donde como método de integración se utilizó Euler con un paso de 0.001 segundos con 10 segundos de simulación para cada individuo. Se limitó el par máximo de 150 Nm a 50 Nm para la articulación 1 y de 15 Nm a 10 Nm para la articulación 2, con el fin de realizar una implementación como trabajo a futuro y evitar la saturación de los controladores.

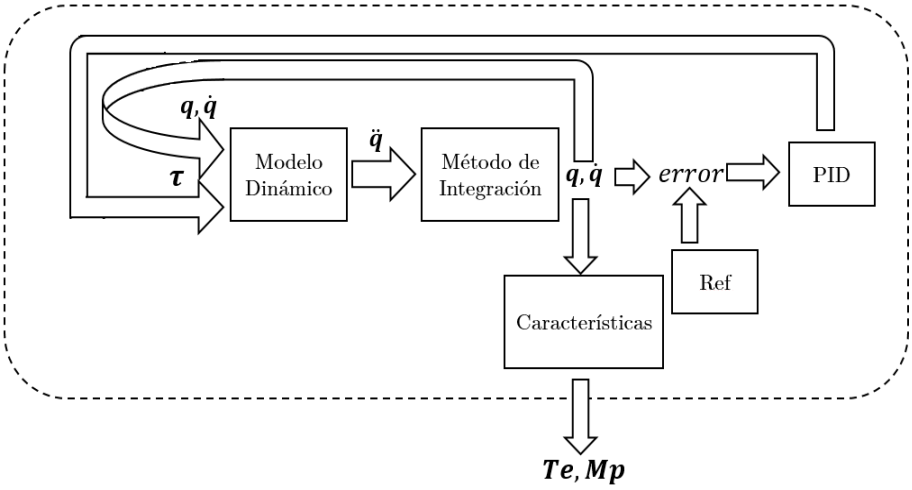


Fig. 4. Diagrama de la función fitness.

3. Resultados

En la Tabla 2 se muestran las ganancias encontradas. Donde se puede observar el porcentaje del sobreimpulso máximo M_p, el tiempo de establecimiento T_e y las ganancias de los controladores. De los 10 individuos finales se muestran 5, se puede apreciar que algunas configuraciones de ganancias disminuyeron el sobreimpulso pero su tiempo de establecimiento aumento. En otros casos el tiempo de establecimiento disminuyó pero aumento el sobreimpulso. Por inspección se elige la mejor configuración de este grupo de ganancias dependiendo de los requerimientos de diseño que convengan al usuario.

Tabla 2. Ganancias encontradas.

M_{p1} %	T_{e1} (seg)	M_{p2} %	T_{e2} (seg)	K_{p1}	K_{i1}	K_{d1}	K_{p2}	K_{i2}	K_{d2}
18,25	5.55	1e-6	0.16	11.07	77.37	8.07	145.1	10.3	5.52
2e-8	3.23	11.76	1.19	31.4	55.91	6.75	89.7	121	3.9
33.32	9.92	1.88	0.15	6.59	83.61	7.78	135.7	6.31	4.48
1.37	1.34	5.34	0.327	8.02	51.73	8.38	76.55	42.2	3.52
11.41	2.32	1.68	0.46	97.97	133.05	31.26	241.83	31.87	31.87

Eligiendo la siguiente configuración de ganancias por sus tiempos de establecimiento y simulando el sistema a 5 segundos con una referencia de 1 radian para ambos controladores:

$$K_{p1} = 97.97 \quad K_{i1} = 133.05 \quad K_{d1} = 31.26$$
$$K_{p2} = 241.83 \quad K_{i2} = 31.87 \quad K_{d2} = 31.87.$$

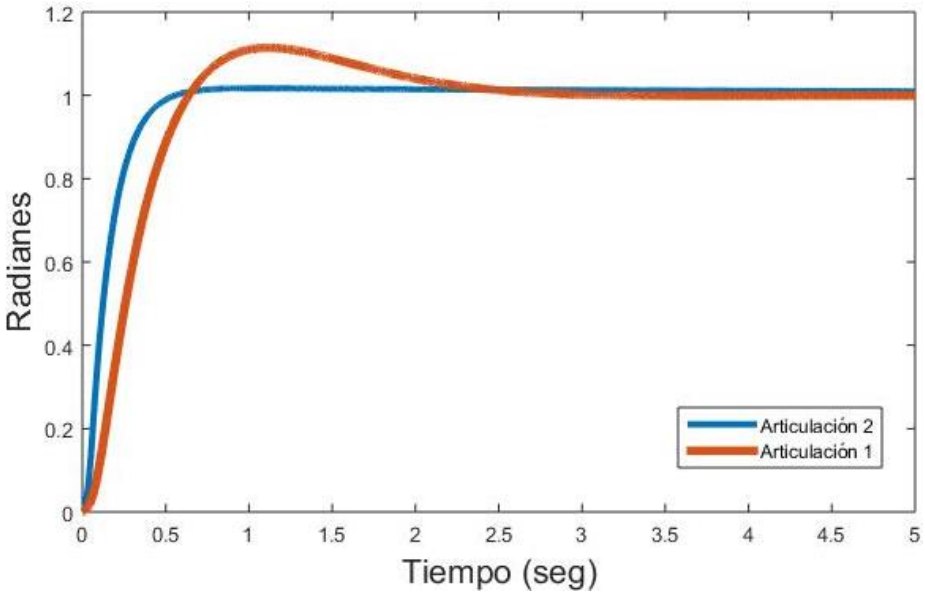


Fig. 5. Respuesta del sistema.

Se graficó la respuesta del sistema, observe que el sobreimpulso máximo (Fig. 5) no supera el 15% y el tiempo de establecimiento no rebasa 2.5 segundos para el primer eslabón. Para el segundo eslabón el sobreimpulso no superó el 2% y el tiempo de establecimiento no rebasa 0.5 segundos.

4. Conclusiones

El uso de los algoritmos genéticos para la optimización es viable debido a que a partir de un espacio de búsqueda grande con ciertas restricciones es posible encontrar la configuración que optimiza el desempeño de la respuesta del sistema, es decir, las ganancias que reducen el tiempo de establecimiento y el sobreimpulso máximo.

Los resultados obtenidos presentan sobreimpulsos menores al 15%, tiempos de establecimiento menores a 2.5 segundos tomando en cuenta que es un sistema MIMO, no lineal y que no se usó el par máximo para ambas articulaciones.

La desventaja de la primera sintonización, es que requiere de un tiempo considerable (1 hora) debido a que el rango de búsqueda, tamaño de la población y número de generaciones son grandes. Una vez que se han encontrado ganancias aceptables se puede reducir el rango de búsqueda. Como trabajo a futuro se

implementará el sistema en un Controlador digital de señales (DSC, por sus siglas en ingles) para tener una mayor aproximación a la implementación con un robot manipulador real.

Agradecimientos. David Bedolla Martínez, con número de becario 577881, agradece al CONACyT por la beca otorgada. Así mismo se agradece al programa Cátedras-CONACYT con el proyecto 621 "Desarrollo de mecanismos robóticos para rehabilitación física", por la participación en éste trabajo.

Referencias

1. Santos, S.O.: Sintonización de un controlador PID basado en un algoritmo heurístico para el control de un Ball and Beam. Tesis, Universidad Autónoma de Querétaro (2014)
2. Åström, K.J., Hägglund, T.: The future of PID control. Control Engineering Practice, No 9, pp. 1163–1165 (2001)
3. PID tuning using extremum seeking: online, model free performance optimization. IEEE Control Systems, No. 79, pp. 70–79, March (2006)
4. Chang, W.D.: A multi-crossover genetic approach to multivariable PID controller tuning. Expert systems with applications, No. 32, pp. 620–626 (2007)
5. Iruthayarajan, M.W., Baskar, S.: Evolutionary Algorithms based design of multivariable PID controller, Expert Systems and applications, no 36, pp. 9159–9167 (2009)
6. Luna-Ortiz, C.: Diseño e implementación de un algoritmo genético en FPGA para sintonización de controladores PID, Tesis, Universidad Autónoma de Querétaro (2011)
7. Oha, S.-K., Janga, H.-J., Pedrycz, W.: The design of a fuzzy cascade controller for ball and beam system: A study in optimization with the use of parallel genetic algorithms. Engineering Aplications of Artificial Intelligence, No. 22, pp. 261–271 (2009)
8. Hultmann Ayala, H.V., dos Santos Coelho, L.: Tuning of PID controller based on a multiobjective genetic algorithm applied to a robotic manipulator. Expert System with Applications, No. 39, pp. 8968–8974 (2012)
9. Kelly, R., Santibáñez, V.: Control de movimiento de Robots Manipuladores. Ed. Pearson Educación (2003)
10. Leyva, F. H. R., Gaspar, L. A. P., Espinosa, F. S.: Simulación de un Robot de Dos Grados de Libertad Usando "Hardware-in-the-loop" con Base en Instrumentación Programable, Pistas educativas (2015)
11. Cortés, F. R.: Robótica, Control de Robots Manipuladores. México D.F.: Alfaomega Grupo Editor, febrero (2013)

Desarrollo de un sistema lógico difuso para el control de la locomoción bípeda de un robot humanoide NAO

Karen Lizbeth Flores-Rodríguez, Felipe Trujillo-Romero

Universidad Tecnológica de la Mixteca,
División de Estudios de Posgrado, Huajuapan de León, Oaxaca,
México

karenflores350@hotmail.com,ftrujillo@mixteco.utm.mx

Resumen. En el siguiente trabajo se desarrolla un sistema lógico difuso para el control de la locomoción bípeda de un robot humanoide NAO. Se considerando como entradas las posiciones de los ángulos de las piernas y como salida la posición cercana a cero del ángulo del torso ambas con respecto a un marco de referencia local independiente. El trabajo parte del modelo *Walking Toy* el cual es un sistema dinámico que describe la locomoción de un bípedo. Se planea evadir la complejidad del sistema dinámico y se propone un sistema difuso de inferencias que permita estabilizar una trayectoria de caminado para el sistema. Los resultados son aceptables, se obtuvo la simulación del caminado estable para el robot NAO.

Palabras clave: Lógica difusa, locomoción bípeda, Walking Toy, robot NAO.

1. Introducción

La investigación sobre locomoción de robots bípedos es el estudio y desarrollo de métodos para lograr estabilidad y equilibrio en el caminado basado en el caminado humano. Este caminado se analiza como una órbita periódica de una fase estable que alterna con una fase inestable. El caminado, por definición, es moverse a un paso moderado levantando y poniendo un pie delante de otro en turnos mientras el otro pie está sobre el suelo. Al menos un pie debe estar tocando el suelo en todo momento para considerarse caminado. El objetivo del análisis de locomoción de robots bípedos es diseñar movimientos de articulaciones y controladores asegurando que este no caiga al caminar. El caminado bípedo se realiza como un conjunto de posiciones de ángulos deseados en el tiempo para las articulaciones de las piernas, conocido también como patrón de caminado. El caminado se considera estable si el único contacto entre el bípedo y el piso es realizado con las plantas de los pies.

El marco básico del control de la locomoción bípeda es el generador de patrones de caminado y la estabilización de los mismos. En este trabajo se

propone un control lógico difuso para la generación de dichos patrones y la estabilizacion de la locomoción bípeda. Se planea evadir la complejidad de los sistemas dinámicos y se propone un sistema difuso de inferencias que permita estabilizar una trayectoria de caminado. Utilizando lógica difusa se proporciona un mecanismo de inferencia que permite simular los procedimientos de razonamiento humano en sistemas basados en el conocimiento. El trabajo parte de las restricciones que se plantean en la aplicación del modelo *Walking Toy* [16], para la locomoción de robots bípedos. Este modelo se describe en una sección posterior del trabajo.

La teoría de la lógica difusa proporciona un marco matemático que permite modelar la incertidumbre de los procesos cognitivos humanos de forma que pueda ser tratable por una computadora. Básicamente la lógica difusa es una lógica multivaluada que permite representar matemáticamente la incertidumbre y la vaguedad, proporcionando herramientas formales para su tratamiento. Se dice que, cualquier problema del mundo puede resolverse como: dado un conjunto de variables de entrada (espacio de entrada), se debe obtener un valor adecuado de variables de salida (espacio de salida). La lógica difusa permite establecer este mapeo de una forma adecuada, atendiendo a criterios de significado (y no de precisión). El documento se organiza como sigue. En la sección 2 se presenta el estado del arte referente principalmente a contribuciones con el robot humanoide NAO. En la sección 3 se describe brevemente el *Control Óptimo Difuso* para la resolución de sistemas dinámicos. en la sección 4 se presenta el *Sistema de Locomoción de Robots Bípedos* donde se explica el modelo *Walking Toy*. En la sección 5 se presenta el *Desarrollo del Controlador Lógico Difuso* que incluye la *Fuzzificación, Inferencia y Defuzzificación*. En la sección 6 se plasman las pruebas y resultados del trabajo. Se muestra el diseño del controlador difuso con el toolbox de MATLAB, la herramienta Simulink y una simulación en Webots. Finalmente, en la sección 7 se dan algunas conclusiones del presente trabajo.

2. Estado del arte

Desde que la plataforma NAO [1] fue elegida como plataforma estándar de la competencia de fútbol robótico RoboCup [11], las investigaciones en el área de locomoción bípeda por parte de universidades y centros de investigación en todo el mundo se incrementaron, pues este robot permite a los investigadores enfocarse en la creación de algoritmos de resolución para los problemas típicos de los robots humanoides y dejar un poco de lado el desarrollo de plataformas robóticas. Entre lo resaltable de la liga de plataforma estándar de RoboCup se encuentran los siguientes trabajos. En [7], se presentan una sólida marcha de circuito cerrado. Basado en el análisis y control del centro de masa soportada por métodos de balanceo. En [4], utilizan el método del péndulo invertido para la generación de movimientos del robot y el criterio del punto de momento cero ZMP para asegurar la buena ejecución de los movimientos. Con una velocidad de 25 cm/seg, es uno de los más rápidos. En [8], presentan una marcha que radica en el uso del modelo del péndulo invertido con duración de fase dinámica para el

modelado de un rápido y robusto caminado omnidireccional, responsivo y reactivo a perturbaciones externas. Con una velocidad de 31 cm/seg hacia adelante y un paso de 7cm, 12 cm/seg de lado con un paso de 8 cm, 22 cm/seg hacia atrás y a 92°/seg cuando rota en su lugar. En [9], se describe una locomoción omnidireccional, con una velocidad hacia adelante de 34 cm/seg. Este caminado implementa la dinámica del péndulo invertido y el criterio de punto de momento cero y el enfoque de descomposición de la marcha en plano dinámico sagital y coronal, para después recombinarlos sincronizadamente.

Otro trabajos relacionados con el robot NAO, como [5], desarrolla un algoritmo sobre locomoción bípeda llamado *Sammy's Walks*. Se basa en estudios de frecuencia, aprovecha las oscilaciones propias del sistema para determinar los ciclos límite y generar la locomoción mas eficiente. Este inspirado en los caminantes pasivos, no entrega el caminado esteticamente estable pero si un gran ahorro energía. Por otro lado, en [14], se presenta una técnica que encuentra automáticamente modos de caminar estables y rápidos. Modelizando la caminata como ondas acopladas reduce el espacio de parámetros y mantiene gran diversidad de movimientos. Otra investigación presentada en [3], describe un enfoque libre de modelo donde utilizan un controlador compuesto por neuronas lineales y no lineales. Se obtiene retroalimentación desde las señales de salida para generar mejores señales y mejorar la estabilidad del caminado. El movimiento de los brazos es utilizado para hacer el caminado suave y robusto. En [10], se introduce el modelo predictivo, el cual es capaz de manejar las restricciones dinámicas que puede presentar el punto de momento cero. Este enfoque es adaptable a cualquier plataforma robótica, como el robot HRP-2 y el robot NAO. En [13], se describe un ciclo abierto de caminado en diferentes pendientes. En el cual la planeación de trayectoria es presentada usando las ecuaciones de semi elipse de movimiento. Este enfoque alcanza la velocidad de caminado de 17 cm/s.

Los métodos de computación flexible como las redes neuronales y la lógica difusa se han vuelto populares en la resolución de dinámicas de sistemas complejas. La lógica difusa ha sido aplicada con éxito para la generación de locomoción para robots. Trabajos como [2], [15], [19], [12], [6] y [17] utilizan lógica difusa para la generación de patrones de caminado y estabilización del mismo. En [18] se describe un controlador omnidireccional de caminado bípedo basado en lógica difusa para el robot NAO. Este controlador se encarga de la restriccion de la longitud de paso e inclinación del cuerpo del robot, tomando como entradas la direccion deseada y ángulo de rotación. El criterio ZMP es utilizado como base para la restricción de la ubicación de la planta del pie del robot.

En resumen, el estudio de la locomoción bípeda puede resolverse mediante diferentes enfoques siendo los más populares el enfoque de péndulo invertido y el criterio ZMP para asegurar la estabilidad. La investigación en esta área radica en las variantes de los enfoques para la resolucíon óptima, mejora de estabilidad y reacción a perturbaciones externas. Se debe considerar que cada robot humanoide implementa el método de resolución para locomoción bípeda que más se adapte a su arquitectura y configuracion. En cuanto a las marchas generadas para el robot NAO, la constante de investigación y desarrollo se enfoca

en mejorar la estabilidad y aumentar la velocidad para fines de fútbol robótico, los algorimtos deben ser rápidos y no deben consumir demasiados recursos pues se deben solucionar otros problemas como navegación, reconocimiento de objetos, comunicación, entre otros.

3. Control óptimo difuso

El control estándar se basa en el conocimiento de ciertas ecuaciones diferenciales $\dot{x} = f(x, u)$ con $x(t_0) = x_0$ que describen la dinámica del sistema. Si no se conoce, entonces se puede usar cierto conocimiento para proponer leyes de control. Por ejemplo:

$$R_j : if\ (x, u)\ is\ A_j\ then\ x\ is\ B_j,\ j = 1, 2, ..., r. \tag{1}$$

A lo anterior se le conoce como control difuso, el cual es un camino para y transformar conocimiento lingüístico en leyes de control. Para esto se considera un proceso de tres pasos:

1. Modelado de conceptos difusos en reglas.
2. Elección de conectividades difusas en reglas.
3. Elección de procesos de defuzzificación.

Considerando un sistema MISO (múltiples entradas una salida) de reglas donde la variable de entrada es: $x = (x_1, x_2, ..., x_n)\ \epsilon X = (X_1, X_2, ..., X_n)$ y la variable de salida es $u = Y$. Primero se especifican particiones difusas de todo el espacio involucrado A_j y B_j en X y Y correspondientes a variables lingüísticas como *small positive*. Después se definen las reglas *IF-THEN* correspondientes a esas particiones difusas de expertos o información de entrenamiento. El procedimiento se define en los siguientes tres pasos:

Paso 1: $R_j : if\ x\ is\ A_j\ then\ u\ is\ B_j,\ j = 1, 2, ..., r$ Las variables lingüísticas A'js y B'js son subconjuntos difusos de espacios apropiados. En el espacio *expertos* para $x_j\ \epsilon X, i = 1, 2, ..., m$ se tienen N expertos con la proposición $x_i\ is\ A$ verdadero. El grado de x_i en A, $A(x_i)$ es M/N donde M es el número de expertos. Las reglas de control deben contener modificadores difusos como *muy* o *casi todo*, etc. En base a la información de los expertos se definen las funciones de membresía y se especifican los parámetros de las funciones.

Paso 2: Después de modelar las etiquetas lingüísticas en reglas como conjuntos difusas de conjuntos apropiados se seleccionan las conectividades lógicas como t-norma (min o producto), t-conorma (max o doble producto) o negación (1-x).

Paso 3: Teniendo un panorama de salidas difusas B como un subconjunto difuso del espacio de salida, se debe resumir en una sola variable $u*$ a usar como control de acción actual. Es necesario elegir un proceso de defuzzificación $D : [0, 1]^U \rightarrow U$. En la Tabla 1 se muestran algunos ejemplos de proceso de defuzzificación.

Tabla 1. Procesos de defuzzificación.

Método	**Función**
Centroide	$u* = \frac{\int_U uB(u)du}{B(u)du}$
Media de la máxima	$u * eselpromediodeloselementosdeU$
Centro de área	$u * eselvalormediodelareatotaldeB$

4. Sistema de locomoción de robots bípedos

Para evitar la complejidad de la dinámica de las piernas del robot humanoide NAO se considera un sistema simple. El sistema parte del enfoque *Walking Toy*[16] el cual se puede apreciar en la Figura 1. Este modelo cuenta con dos piernas de masa M_l y longitud idéntica L, conectadas a una rueda (cuerpo) inercial de masa M_W mediante dos motores rotatorios que provee el torque necesario para mover las piernas. Ambas piernas se encuantran conectadas al eje central del cuerpo rodante, el cual se considera la cadera de la máquina. El modelo matemático para este sistema se presenta en (2).

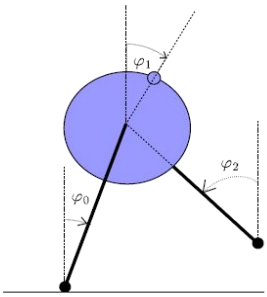


Fig. 1. Walking Toy esquemático.

$$\begin{aligned} J_0\ddot{\varphi}_0 &= mgL\sin\varphi_0 - \tau_1, \\ J_1\ddot{\varphi}_1 &= \tau_1 - \tau_2, \\ J_2\ddot{\varphi}_2 &= \tau_2, \end{aligned} \tag{2}$$

donde J_0 es la inercia de la pierna apoyada y el cuerpo con respecto al punto pivote del suelo. J_1 y J_2 representan la rotación inercial del cuerpo y la pierna balanceada con respecto a la cadera. Los torques aplicados, actuando como control son denotados por τ_1 y τ_2. Sabiendo que $m = M_l + M_W$. Cuando la pierna balanceada aterriza en el suelo, el modelo del sistema cambia los roles de las piernas.

Del modelo esquemático del sistema, es evidente la existencia de ciertas restricciones que deben ser consideradas para la realización de un caminado bípedo:

1. El ángulo del pie apoyado sobre el suelo debe ser pequeño y no debe pasar de cierto rango, $|\varphi_0| < \frac{\pi}{2}$,
2. La pierna balanceada nunca tendrá la punta extrema por debajo del nivel del suelo horizontal, siempre será positiva $|\varphi_0(t)| \leq |\varphi_2(t)| \forall t$.

Además de las restricciones antes mencionadas, se conocen ciertas características del sistema. Teniendo en cuenta que el objetivo es que el sistema ejecute un paso suavemente desde una posición de reposo del tiempo $t = t_0$ al tiempo $t = T > t_0$. Se Considera como posición de reposo aquella en la cual el ángulo de la pierna apoyada contra el suelo φ_0 es simétrica al ángulo de la pierna balanceada φ_2 (3):

$$\varphi_0 = -\varphi_2. \tag{3}$$

Teniendo en cuenta que el sistema considera los torques de los eslabones de las piernas como entradas de control, se define la trayectoria en base a los ángulos de las piernas del bípedo. La posición inicial de pie es determinada por el ángulo inicial arbitrario de la pierna fija. Este debe ser un ángulo positivo (4):

$$\varphi_0(t_0) = \alpha. \tag{4}$$

Al final del primer paso el bípedo debe alcanzar una nueva posición de pie caracterizada por la pierna balanceada (5):

$$\varphi_0(T) = -\beta,\ \beta > 0. \tag{5}$$

La posición inicial de los pies al tiempo t_0 es entonces (6):

$$\begin{gathered} \varphi_0(t_0) = -\alpha, \\ \varphi_2(t_0) = \alpha, \\ \varphi_1(t_0) = \frac{J_0}{J_1}(1 - \frac{J_2}{J_0})\alpha. \end{gathered} \tag{6}$$

Similarmente al tiempo $t = T > 0$, tiempo en completarse el primer paso, el estado del sistema debe satisfacer la siguiente relación (7):

$$\begin{gathered} \varphi_0(T) = \beta, \\ \varphi_2(T) = -\beta, \\ \varphi_1(T) = -\frac{J_0}{J_1}(1 - \frac{J_2}{J_0})\beta. \end{gathered} \tag{7}$$

Una vez que se complete el primer paso en el intervalo de tiempo $[t_0, T]$ es necesario permutar la pierna apoyada al suelo con la pierna balanceada al tiempo $t = T$. De esta manera bastará con tomar el mismo modelo pero cambiando los valores de φ_0* y φ_2*. Entonces se puede comenzar en el tiempo T un nuevo paso con el mismo modelo al estado estable con $\varphi_0 = \beta$ y $\varphi_2 = -\beta$ (8):

$$\begin{gathered} J_0\ddot{\varphi_0} = mgL\sin\varphi_0 - \tau_2, \\ J_1\ddot{\varphi_1} = \tau_2 - \tau_1, \\ J_2\ddot{\varphi_2} = \tau_1. \end{gathered} \tag{8}$$

Las condiciones finales de reposo del primer paso se convierten en las condiciones iniciales del segundo paso, las cuales pueden ser de longitud arbitraria. Se considera el intervalo $[T = T_1, T_2]$.

5. Desarrollo del controlador lógico difuso

5.1. Fuzzificación

La *Fuzzificación* considera dos variables de entrada para el sistema de locomoción bípeda: el ángulo de la pierna apoyada al suelo y el ángulo de la pierna balanceada. Estos ángulos se intercalan conforme se ejecuta el caminado, por comodidad se define *ángulo 1* como pierna derecha y *ángulo 2* como pierna izquierda. Tomando en cuenta las restricciones mencionadas en la sección de *Locomoción de Robots Bípedos*, se considera que ambos ángulos de las piernas no pueden tener valores mayores al rango -45° a 45°, otra restricción es que los ángulos solo serán iguales en 0° que es la posición de pie. Con estas restricciones se procede a modelar las funciones de membresía para la variable de salida que es definida como el ángulo de inclinación de un punto en el torso del bípedo que debe ser cercano a cero.

Para cada una de las piernas se consideran cinco funciones de membresía de tipo Gaussiana ya que que el caminado bípedo es un ciclo continuo y suave. Las funciones de membresía se definen entre -45°, -20°, 0°, 20° y 45°. Estas son las posiciones del pie: *atrás, menos atrás, centro, menos adelante y adelante*. Las funciones Gaussianas tienen diferentes amplitudes, estas se definen a continuación:

$$x_1, x_2 = \begin{cases} atras & e^{-\frac{1}{2}(\frac{x-(-45)}{12})^2} \\ menos-atras & e^{-\frac{1}{2}(\frac{x-(-20)}{5})^2} \\ centro & e^{-\frac{1}{2}(\frac{x-(0)}{5})^2} \\ menos-adelante & e^{-\frac{1}{2}(\frac{x-(20)}{5})^2} \\ adelante & e^{-\frac{1}{2}(\frac{x-(45)}{12})^2} \end{cases} \tag{9}$$

Para la variable de salida que es el ángulo del torso con respecto a un punto de referencia, se definen tres funciones triangulares. Estas funciones son:

$$y = \begin{cases} menos-centro & \frac{x-(-20)}{(-10)-(-20)}, \frac{20-x}{20-(-10)} \\ centro & \frac{x-(-20)}{(0)-(-20)}, \frac{20-x}{20-(0)} \\ mas-centro & \frac{x-(-20)}{(10)-(-20)}, \frac{20-x}{20-(10)} \end{cases} \tag{10}$$

En base a las variables de entrada se define el grado de pertenencia a las funciones de la variable de salida.

5.2. Inferencia

La inferencia se basa en el conocimiento de expertos mencionado en la sección *Locomoción de Robots Bípedos*, resultado del mismo análisis de la órbita del caminado. En el caminado, ambas piernas nunca tendrán la misma posición más que en la posición de inicio o reposo. En la Tabla 2 se muestran las posibles posiciones de las piernas de un caminado. En verde se muestran las posiciones posibles y en rojo las posiciones restringidas. En base a estas posiciones se definen las reglas difusas mediante proposiciones de Mamdani. Se busca compensar la posición del ángulo del torso para que no se aleje de cero. En color verde obscuro se muestran las posiciones de las piernas en las que la posición del ángulo del torso debe ser cero. En color verde claro se muestran las posiciones de las piernas en las que el ángulo debe compensarse para acercarlo a cero.

Derecho	Adelante-Atrás	Adelante-Menos Atrás	Adelante-Centro	Adelante- Menos Adelante	Adelante-Adelante
	Menos Adelante-Atrás	Menos Adelante-Menos Atrás	Menos Adelante-Centro	Menos Adelante-Menos Adelante	Menos Adelante-Adelante
	Centro-Atrás	Centro- Menos Atrás	Centro-Centro	Centro- Menos Adelante	Centro-Adelante
	Menos Atrás-Atrás	Menos Atrás- Menos Atrás	Menos Atrás-Centro	Menos Atrás-Menos Adelante	Menos Atrás-Adelante
	Atrás-Atrás	Atrás- Menos Atrás	Atrás-Centro	Atrás- Menos Adelante	Atrás-Adelante
	Izquierdo				

Fig. 2. Posición de las piernas en los pasos de un caminado.

En base a la Tabla 2 se establecieron 18 reglas difusas, de las cuales se muestran a continuación las nueve reglas pertenecientes a las sombreadas de color verde obscuro:

- IF (ángulo 1) is (centro) AND (ángulo 2) is (adelante) THEN (ángulo 3) is (centro)
- IF (ángulo 1) is (centro) AND (ángulo 2) is (centro) THEN (ángulo 3) is (centro)
- IF (ángulo 1) is (centro) AND (ángulo 2) is (atrás) THEN (ángulo 3) is (centro)
- IF (ángulo 1) is (adelante) AND (ángulo 2) is (centro) THEN (ángulo 3) is (centro)
- IF (ángulo 1) is (menos adelante) AND (ángulo 2) is (centro) THEN (ángulo 3) is (menos centro)
- IF (ángulo 1) is (menos atrás) AND (ángulo 2) is (centro) THEN (ángulo 3) is (mas centro)
- IF (ángulo 1) is (atrás) AND (ángulo 2) is (centro) THEN (ángulo 3) is (centro)
- IF (ángulo 1) is (adelante) AND (ángulo 2) is (atrás) THEN (ángulo 3) is (centro)
- IF (ángulo 1) is (atrás) AND (ángulo 2) is (adelante) THEN (ángulo 3) is (centro)

5.3. Defuzzificación

Para la defuzzificación se eligió el proceso de centroide, el cual es el promedio de todo el espacio de salida y, como se mencionó en la sección Control Óptimo Difuso, el método queda definido como en 11:

$$y = \frac{\sum_{i=1}^{N} y_i \mu(y_i)}{\sum_{i=1}^{N} \mu(y_i)},\ i = 1, 2, ..., n, \tag{11}$$

donde y será la salida única, $\mu(y_i)$ es el valor de pertenencia de y_i en la función de membresía de la variable de salida.

6. Pruebas y resultados

Se hizo uso del toolbox de lógica difusa en MATLAB para el diseño del conrolador difuso del sistema definido en las secciones anteriores. Para demostrar el funcionamiento del control difuso se utilizó la herramienta Simulink de MATLAB y se realizó una simulación de en Webots con el robot humanoide NAO. En las siguientes secciones se ilustra el procedimiento realizado para el diseño de controlador difuso, la aplicación en Simulink y la simulación en Webots.

6.1. Diseño del controlador difuso

El diseño del controlador difuso se basa en lo definido en la sección de Desarrollo. En la Figura 3, se muestra el diagrama de bloques del controlador difuso para el sistema basado en las restricciones del modelo *Walking Toy*. Este controlador cuenta con dos entradas *angulo*1 y *angulo*2 para cada una de las piernas del bípedo. La salida del sistema es *angulo*3 el cual es el ángulo del torso con respecto a un punto de referencia que debe ser cercano a cero.

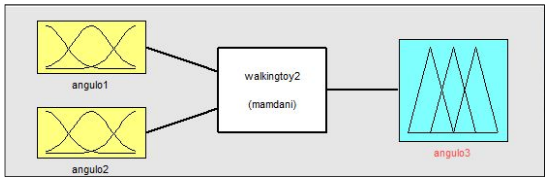


Fig. 3. Diagrama de bloques del controlador difuso para el sistema.

En la Figura 4 se muestran las funciones de membresía de las variables de entrada del sistema. Estas variables de entrada son los ángulos de las piernas que representan las posiciones de estas en el caminado. Cada variable cuenta con cinco funciones Gaussianas que fueron definidas con anterioridad. Estas funciones se definieron en un rango de -45° a 45° la cual se consideran los grados

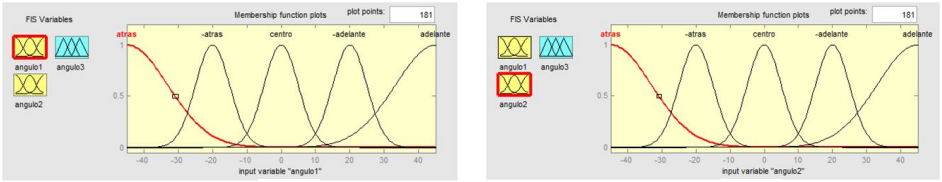


Fig. 4. Muestra las funciones de membresía de las variables de entrada del sistema. (a) Variable ángulo 1, (b) Variable ángulo 2.

que pueden tomar las piernas. Las funciones son *atrás, menos atrás, centro, menos adelante y adelante.*

En la Figura 5 se muestran las funciones de membresía de la variable de salida. Las funciones son triangulares y están definidas entre -20° a 20°. En este rango se encuentran los valores permitidos que puede tomar el ángulo del torso con respecto a un punto de referencia para mantenerlo cerca a cero. Estas funciones son menos centro, centro y mas centro. Los centro con modificadores lingüísticos se definen para poder compensar el sistema si este presenta mayor inclinación para cierto lado.

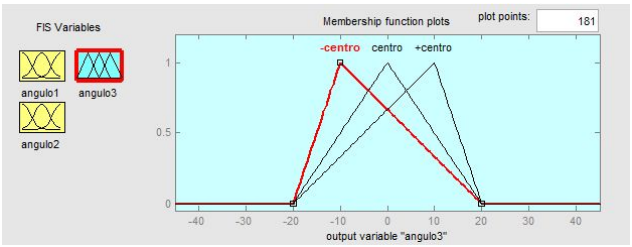


Fig. 5. Funciones de Membresía para la variable de salida del sistema.

Las reglas de inferencia del sistema se definen mediante proposiciones de *Mamdani.* Estas reglas buscan la aproximación al centro definido con el ángulo del torso. Además, buscan la compensación del torso al momento de efectuar pasos.

6.2. Aplicación en Simulink

El sistema se implemento en Simulink utilizando el toolbox de Lógica Difusa. Se definieron dos funciones sinusoidales con amplitud contraria para simular el ciclo del caminado para ambsa piernas. En la Figura 6 se muestra el diagrama de bloques de Simulink del sistema. En el diagrama de bloques del sistema, se

generan dos señales sinusoidales mediante el bloque *Signal Generator*, a cada uno se le agrega un offset de 0.5 para variar un poco la señal. Ambas señales se muestran en una gráfica con el bloque *Scope*. Estas señales entran al bloque *Fuzzy Logic Controller* el cual es el controlador difuso diseñado para el sistema con la aplicación de Lógica Difusa en MATLAB. La variable de salida se muestra en otra gráfica. En las Figuras 7 (a) y 7 (b) se muestran las señales de entrada sinusoidales y la señal de salida cercana a cero para el sistema del controlador difuso.

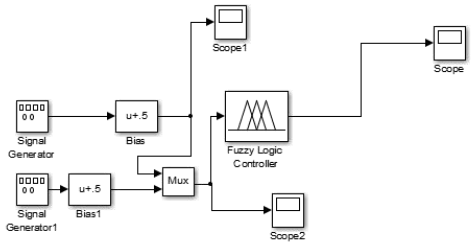


Fig. 6. Diagrama de bloques de Simulink del sistema.

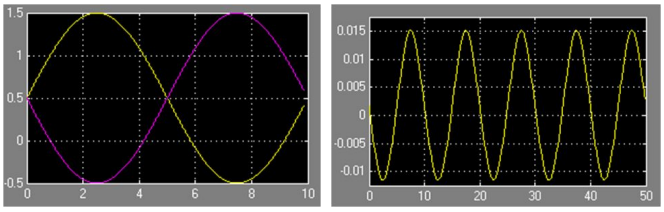


Fig. 7. (a) Variables de entrada representada en ciclos sinusoidales, (b) Señal de salida sinusoidal cercana a cero.

6.3. Simulación

La simulación del sistema lógico difuso para el control de la locomoción bípeda de un robot humanoide NAO se visualizó con el software Webots [1]. Este software es mundo virtual que permite lanzar una simulación de movimientos de NAO tomando en cuenta leyes físicas. Ofrece un lugar seguro para probar el funcionamiento de los comportamientos antes de reproducirlos en un robot

real. Para realizar la simulación se recuperaron los ángulos de las trayectorias obtenidas con el controlador lógico difuso. Los ángulos de las piernas se utilizan para obtener la trayectoria lineal mediante una simple función trigonométrica:

$$x = L \sin \varphi_2, \tag{12}$$

donde x será la posición a colocar el pie del robot en el eje x, L es la longitud de la pierna del robot, (se considera una longitud máxima de 200mm) y φ_2 es el ángulo de la pierna balanceada. La elevación de la pierna se considera de 50mm. Mientras que el ángulo del torso se compensa después de cada paso tomando el valor directamente del controlador difuso y pasandolo al robot. De manera cuantitativa se muestra en la Figura 8 la ejecución de un paso de 5 cm y una zancada de 10 cm por el método desarrollado en este trabajo y el control de Aldebaran. De esta comparativa lo que se puede resaltar es la similitud de estabilidad al avanzar el mismo tramo. Sin embargo, el control sobre la estabilidad del caminado utilizando el algoritmo de Aldebaran no es muy confiable en tramos más extensos, por este motivo se propone el uso del control lógico difuso. En la Figura 9 se muestra la secuencia de imágenes que representan un paso completo con ambas piernas realizado por el robot NAO. en la secuencia se observa que el torso se compensa al finalizar un paso con cada pierna.

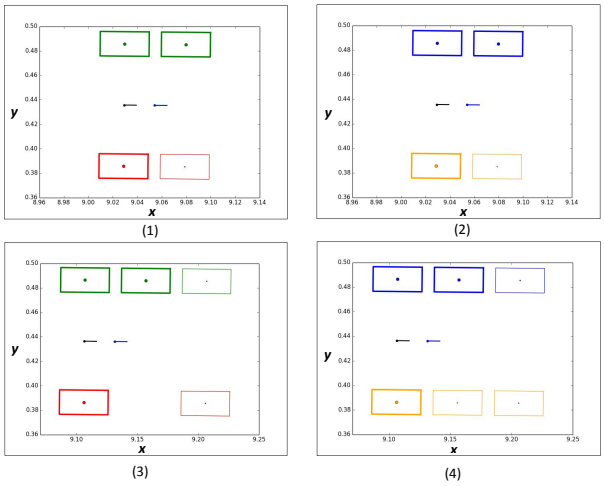


Fig. 8. Secuencia del caminado capturando la posición de los pies. (1) Un paso realizado mediante parámetros del controlador difuso. (2) Un paso realizado mediante el algoritmo de Aldebaran. (3) Una zancada realizada mediante parámetros del controlador difuso. (2) Una zancada realizada mediante el algoritmo de Aldebaran.

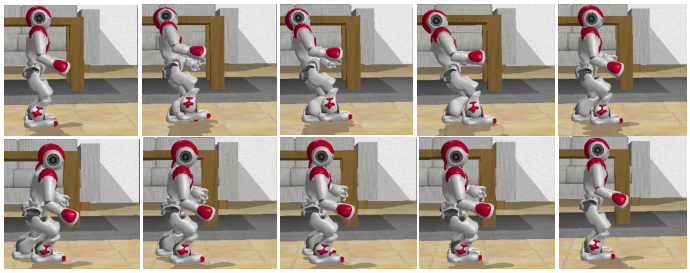


Fig. 9. Secuencia del caminado simulado en Webots por el robot NAO, se muestra un paso con el pie derecho y un paso con el pie izquierdo.

7. Conclusiones

Los sistemas de control convencionales son precisos y bien calculados para modelos matemáticos bien definidos. En sistemas donde no se conoce o no se puede modelar el sistema dinámico el control mediante lógica difusa es mejor opción. La razón por la que la lógica difusa será suficiente para poder controlar un sistema es porque el conocimiento que se utiliza es deducido por experiencia, observaciones y aplicaciones de los mismos. El razonamiento humano es difícil de modelar matemáticamente, este razonamiento se utiliza en la vida diaria y en cualquier evento cotidiano que describa una o unas acciones de un problema.

El sistema Walking Toy es un sistema dinámico que describe un modelo no lineal. Este sistema ha sido resuelto mediante control convencional con técnicas de control no lineal, sin embargo, no se conoce aplicación real. La resolución realizada en este trabajo fue una aproximación al control del caminado planteado por el método convencional. Los resultados obtenidos mostrados cumplieron con el objetivo de controlar el torso del bípedo para que se mantuviera cercano a la referencia cero. Como ventajas se tiene que el controlador difuso se implementa sin tanta complejidad. Fueron suficientes pocas reglas difusas para obtener un modelo robusto de estabilización del torso. El sistema es capaz de lidear con situaciones no previstas en la modelación. Entre las limitantes que tiene el sistema se puede mencionar la implementación. Esto debido a que solo se evalua con trayectorias rectas en un ambiente de simulación.

Como trabajos futuros se proponen la implementación del método convencional y lógica difusa con este sistema aplicado al robot humanoide NAO en una plataforma real. El objetivo de la implementación es la realización de una comparativa entre ambos controladores para verificar cuál de estos tiene mejores resultados en desempeño. Además de la implementación con trayectorias curvas y pendientes.

Referencias

1. ALDEBARAN, R.: Nao documentation (2015)

2. Ankarali, A.: Fuzzy logic velocity control of a biped robot locomotion and simulation. Advanced Robotic Systems 9 (2012)
3. Azizi, V., Haghighat, A.T.: Fast and robust biped walking involving arm swing and control of inertia based on neural network whith harmony search optimizer. Methods and Models in Automation and Robotics (MMAR) (2011)
4. Czarmetzki, S., Jochmann, G., Kerner, S.: Nao devils dormound team description for robocup 2010. Tech. rep., Robotics Reserch Institute Section Information Technology TU Dormound University (2010)
5. Fernández-Iglesias, S.: Locomoción bípeda del robot humanoide nao. Tech. rep., Universidad Politécnica de Catalunya (November 2009)
6. Gorrostieta, E., Vargas-Soto, E.: Fuzzy algorithm of free locomotion for a six legged walking robot. Computación y Sistemas 11(3), 260–287 (2008)
7. Graf, C., Härtl, A., Rö, T., Laue, T.: A robust closed-loop gait for the standard platform league humanoid. In: Humanoid Soccer Robots. pp. 30–37. IEEE-RAS Intl. Conf On Humanoid Robots (December 2009)
8. Graf, C., Röfer, T.: A center of mass observing 3d-lipm gait for the robocup standard platform league humanoid. In: RoboCup 2011: Robot Soccer World Cup XV. pp. 102–113. Springer Berlin Heidelberg (2013)
9. Hengst, B.: Runswift walk2014 report robocup standard platform league. Tech. rep., School of Computer Science and Engineering University of New South Wale (2014)
10. Herdt, A.: Model predictive control of a humanoid robot. Ph.D. thesis, Insitut des sciences et technologies (2012)
11. Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E.: Robocup: the robot world cup initiative (1997)
12. Liu, Y., Wensing, P.M., Orin, D.E., Schmiedeler, J.P.: Fuzzy controlled hopping in a biped robot. Robotics and Automation (ICRA) (2011)
13. Massah, A., Sharifi, A., Salehinia, Y., Najafi, F.: An open loop walking on different slopes for nao humanoid robot. Proceedings Engineering pp. 296–304 (2012)
14. Rivas, F.M., Cañas, J.M., González, J.: Aprendizaje automático de modos de caminar para un robot humanoide. ResearchGate (2011)
15. Seven, U., Akbas, T., Fidan, K., Erbatur, K.: Bipedal robot walking control on inclined planes by fuzzy reference trajectory modification. Soft Comput (2012)
16. Sira-Ramírez, H., Agrawal, S.: The walking toy. In: Differentially Flat Systems, pp. 307–320. Marcel Dekker (April 2004)
17. Vargas-Soto, E., Gorrostieta, E., Sotomayor-Olmedo, A., Ramos-Arreguin, J., Tovar-Arriaga, S.: Design of fuzzy algorithms locomotion for six legged walking robot. International Journal of Physical Sciences 7(11), 1811–1819 (March 2012)
18. Yanjun, Z.: Fuzzy omnidirectional walking controller for the humanoid soccer robot. The 4th Workshop on humanoid Soccer Robots (2009)
19. Zaidi, A., Rokbani, N., Alimi, A.M.: A hierarchical fuzzy controller for a biped robot. Individual and Collective Behaviors in Robotics (ICBR) (2013)

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
septiembre de 2016
Printing 500 / Edición 500 ejemplares