INNOVACIONES DE LAS CIENCIAS COMPUTACIONALES EN SISTEMAS INTELIGENTES Y CIBERSEGURIDAD

María del Carmen Santiago Díaz

INNOVACIONES DE LAS CIENCIAS COMPUTACIONALES EN SISTEMAS INTELIGENTES Y CIBERSEGURIDAD

María del Carmen Santiago Díaz

Gustavo Trinidad Rubín Linares

Yeiny Romero Hernández

Ana Claudia Zenteno Vázquez

Judith Pérez Marcial

(Editores**)**

María del Carmen Santiago Díaz

(Coordinadora)

María del Carmen Santiago Díaz, Gustavo Trinidad Rubín Linares, Yeiny Romero Hernández, Ana Claudia Zenteno Vázquez, Judith Pérez Marcial
(editores BUAP)

María del Carmen Santiago Díaz
(coordinadora BUAP)

María del Carmen Santiago Díaz, Ana Claudia Zenteno Vázquez, Judith Pérez Marcial, Gustavo Trindad Rubín Linares, Roberto Contreras Juárez, Alba Maribel Sánchez Gálvez, Juan Pablo Ucán Pech, Heidy Marisol Marín Castro, Abelardo Gómez Andrade, Raúl Antonio Aguilar Vera, Paola Eunice Rivera Salas, Pedro García Juárez, María Concepción Landa Arnaiz, Osslan Osiris Vergara, Jéssica Nayeli López Espejel, José de Jesús Lavalle Martínez, Carina Toxqui Quitl, Luis Enrique Colmenares Guillén, Hermes Moreno Álvarez, José Andrés Vázquez Flores, Francisco Marroquín González, Miguel Morales Sandoval, Bárbara Emma Sánchez Rinza, Abel Alejandro Rubín Alvarado, Mario Rossainz López, Oleg Starostenko Basarab, Pedro Bello López, Guillermo De Ita Luna, Mariano Larios Gómez.
(revisores)

Primera edición: 2023

ISBN: 978-607-8857-79-1

Montiel & Soriano Editores S.A. de C.V.

15 sur 1103-6 Col. Santiago Puebla, Pue.

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

Rectora:
Dra. María Lilia Cedillo Ramírez

Secretario General:
Mtra. José Manuel Alonso Orozco

Vicerrector de Investigación y Estudios de Posgrado:
Dr. Ygnacio Martínez Laguna

Directora de la Facultad de Ciencias de la Computación:
M.I. María del Consuelo Molina García

Contenido

Prefacio

La inteligencia Artificial se encuentra en un crecimiento tan acelerado que cuesta trabajo imaginarnos todo el potencial que tiene y lo que se está desarrollando para los siguientes años, como el chatgpt, que dicho sirva de paso mucha gente hace unos años dudaba que se podría tener un chat que interactuara con humanos y que además le sirviera no solo para entablar conversaciones inteligentes, sino también generarle documentos que le ayuden en sus tareas cotidianas y que no solo sean las típicas que le ayuden en las actividades escolares de niños y adolescentes, como responder preguntas de geografía, matemáticas, etc. lo cual realmente constituye una gran ayuda porque en nuestra sociedad, en lo general, aunque se tenga un título universitario nunca ha sido garantía de eficiente preparación, pero para finalizar con el chatgpt, en este momento se cuenta con una versión que nos permite generar presentaciones profesionales, edición de imagen y video, etc., cosas realmente increíbles y que aunque se encuentra en la cima del desarrollo de la inteligencia artificial, la competencia es muy fuerte y mientras escribimos esta líneas Google está realizando el lanzamiento de BARD que aunque son herramientas diferentes no dejan de ser una competencia. Pero el desarrollo de estos chats no ha sido trabajo de un par de meses, se sabe que ha llevado varios años, y otros más liberarlo, no solo por aquellas ideas románticas de que pueden destruir el mundo, sino porque se deben limitar las aplicaciones a fin de no crear confusión y mal uso e interpretación de la información que genera, lo cual claramente nos marca una total complicidad con la Ciencia de Datos y la Ciberseguridad, y es que no se conciben tecnologías que estén libres de estos paradigmas, es claro que manejar 170 billones de parámetros y atender a casi 200 millones de usuarios al mes no se puede hacer sin estas 3 tecnologías. Pero afortunadamente todos estos cambios tecnológicos se dan ante nuestros ojos y sentidos, por lo cual ahora requerimos de un acelerado aprendizaje, así es, las tecnologías que nos ponen al alcance no solo requieren sino necesitan de una capacitación para utilizarlas en su máximo potencial a fin de realmente obtener los frutos para lo que han sido diseñadas, y es en este punto que no hay esfuerzos vanos, todos los esfuerzos deben orientarse en aplicar las nuevas tecnologías, desarrollarlas, difundirlas y sobre todo ponerlas incluso en una encrucijada cuestionándolas, para eso son, y es que para nosotros como usuarios de estas tecnologías y parte de una comunidad académica y científica, es una responsabilidad mayor pues se espera que nuestra visión objetiva oriente y potencialice estas herramientas para nuestros estudiantes y en general a la sociedad que los utilizará.

Aunque hemos hablado en estas líneas del tan acertado desarrollo realizado por OpenIA del chatgpt, y en donde los reflectores se encuentran ahora, sobre la IA aún hay muchísimo más, como aplicaciones de IA en medicina, en bienes y servicios, en la industria, en la educación, etc. realmente está en todos lados, no se puede imaginar un sector donde no se encuentre la Inteligencia Artificial y la Seguridad Informática, por lo que este libro es nuestra aportación y en el cual hemos puesto un especial cuidado de los trabajos que se presentan buscando la calidad que se requiere a fin de que sean referencia de otras aportaciones y fuente de conocimiento que despierte la creatividad de estudiantes e investigadores.

María del Carmen Santiago Díaz
Gustavo Trinidad Rubín Linares

Ruido laplaciano y redes Bayesianas en la generación de datos sintéticos

Laplacian noise and Bayesian network in synthetic data generation

Oscar Rene Salgado Guzman, María de Lourdes Sandoval Solís, Marcela Rivera Martínez, Luis René Marcial Castillo
Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, Puebla, México
oscar.salgadog@alumno.buap.mx, maria.sandoval@correo.buap.mx, marcela.rivera@correo.buap.mx, luis.marcial@correo.buap.mx

Abstract

Laplacian's noise and Bayesian networks are used to generate synthetic data when data confidentiality is desired. In this paper, DataSynthesizer [4] software is used to generate synthetic data using Bayesian networks that preserve the correlation matrix and to preserve data privacy, a Laplacian noise is applied. It is shown that for the Adult Data Set [10], when the Laplacian noise is higher, it is observed that the distribution of the attributes is preserved, as well as their confidentiality. While when the degree of the Bayesian network is changed, the correlation matrix is modified.

Resumen

Cuando se desea conservar la confidencialidad de datos se usa ruido laplaciano y redes bayesianas para generar datos sintéticos. En el presente trabajo, se usa el software DataSynthesizer [4] para generar datos sintéticos usando redes bayesianas que conservan la matriz de correlación, para conservar la privacidad de los datos se aplica un ruido laplaciano. En este trabajo, se muestra que para la base de datos Adult Data Set [10], cuando el ruido laplaciano es mayor, se observa que se conserva la distribución de los atributos, además de la confidencialidad de estos. Mientras que cuando se cambia el grado de la red bayesiana se modifica la matriz de correlación.

Keywords and phrases: Generación de Datos Sintéticos, DataSynthesizer, Ruido de Laplace, Redes Bayesianas, Inteligencia Artificial.

1 Introducción

En Inteligencia Artificial se requieren entrenar, por ejemplo, Redes Neuronales Supervisadas, para ello se necesita información, en casos, como en el área de la salud se cuenta con pocos datos que además son confidenciales; o, en el área financiera que se requiere la confidencialidad de los datos. En estos casos, los datos sintéticos se generan y permiten trabajar la base de datos sin comprometer información confidencial [1][2].

Para sustituir los datos históricos reales con el fin de entrenar modelos de inteligencia artificial se generan datos sintéticos de forma artificial. Se recurre a este tipo de técnicas porque los datos reales son insuficientes e imposibilitan el uso de técnicas de inteligencia artificial. Los datos sintéticos permiten conservar la privacidad. [3]

En este documento se utilizan redes bayesianas para generar datos sintéticos, usando la matriz de correlación, las cuales nos permiten identificar el grado de dependencia entre las variables, las redes bayesianas permiten predecir datos, ya que una red bayesiana es mejor que una red neuronal para la inferencia estocástica.

En las siguientes secciones, se presentan los antecedentes del tema, un resumen breve sobre la teoría de las redes bayesianas y ruido laplaciano, para después introducir la biblioteca de Datasynthesizer. Más adelante se exponen los experimentos tanto en la base de datos original, Adult Data Set [10], así como con los datos sintéticos generados. Para posteriormente, presentar un análisis de los resultados y finalizar con las conclusiones. Además, se presentan las referencias de los artículos consultados.

2 Antecedentes

De acuerdo con la revisión bibliográfica que se realizó en agosto de 2022 se obtuvo la siguiente información mostrada en la tabla 1. Observe que los trabajos son de 2020 y 2021. En este trabajo se presenta el estudio de la biblioteca DataSynthesizer.

3 Redes Bayesianas

3.1 Teoría de probabilidad

A continuación, se presentarán algunos conceptos de probabilidad para entender la regla de bayes

Estadística Bayesiana:

Conjunto de herramientas que se utiliza en un tipo especial de inferencia estadística que se aplica en el análisis de datos experimentales en muchas situaciones prácticas de ciencia e ingeniería.

Regla de bayes:

Si los eventos Y_1,, Y_2,, …, Y_k, representan una partición del espacio muestral S, donde P(Y_i,) ≠ 0 para i = 1, 2, …, k, entonces, para cualquier evento X en S, tal que P(X) ≠ 0.

$$P(Y_r|X) = \frac{P(Y_r \cap X)}{\sum_{i=1}^{k} P(Y_i \cap X)} = \frac{P(Y_r)P(X|Y_r)}{\sum_{i=1}^{k} P(Y_i)P(X|Y_i)} \; para\; r = 1, 2, \dots, k. \quad (1)$$

La regla de Bayes es un método estadístico que también es llamado método bayesiano y ha adquirido muchas aplicaciones. En el capítulo 18 de [5] se realiza una introducción al método bayesiano.

Tabla 1: Software generador de datos sintéticos.

Nombre de software	Descripción	Año	Autor	Artículo	Implementación
DataSynthesizer	Captura la estructura de correlación subyacente entre los diferentes atributos mediante la construcción de una red bayesiana.	2021	Fida K. Dankar and Mahmoud Ibrahim	Fake It Till You Make It: Guidelines for Effective Synthetic Data Generation	
Synthetic Data Vault (SDV)	Modela la función de distribución acumulativa F de la población a partir de la muestra.				
Synthpop paramétrico	Genera el conjunto de datos sintéticos secuencialmente un atributo a la vez mediante la estimación de distribuciones condicionales. Utiliza el algoritmo no paramétrico CART (Classification and Regression Trees)				
Synthpop no paramétrico	Utiliza la regresión logística junto con la regresión lineal para generar las distribuciones condicionales.	2021	Fida K. Dankar and Mahmoud Ibrahim	Fake It Till You Make It: Guidelines for Effective Synthetic Data Generation	
Técnicas CART	Método alternativo de síntesis de datos mediante una técnica no paramétrica basada en árboles que utiliza árboles de clasificación y regresión.	2020	Debbie Rankin, BSc, PhD	Reliability of Supervised Machine Learning Using Synthetic Data in Health Care: Model to Preserve Privacy for Data Sharing.	Paquete en R Synthpop
Paramétricas	Asigna métodos paramétricos por defecto a las variables que se van a sintetizar en función de sus tipos.				Synthpop paramétrico
Redes bayesianas	Es un modelo gráfico probabilístico para representar el conocimiento sobre un dominio incierto en el que cada nodo corresponde a una variable aleatoria.				Python DataSynthesizer

3.2 Redes bayesianas

Para entender lo que es una red bayesiana es necesario comprender algunos conceptos de la teoría de grafos. Las siguientes definiciones nos aportarán lo básico para entender el concepto de red bayesiana.

Las redes bayesianas son una representación gráfica de dependencias probabilísticas, en la cual los nodos (X) representan variables aleatorias y los arcos (A) representan relaciones de dependencia directa entre las variables. Con la inferencia bayesiana se puede estimar la probabilidad posterior de las variables no conocidas, en base a las variables conocidas.

Definición: Una red Bayesiana es una 4-tupla (G, f_x, Q, Θ), que representa una distribución de Probabilidad Conjunta donde:

- (G, f_x, Q) es una red causal.
- G es un dígrafo acíclico.
- El conjunto X de nodos de G es un conjunto $\{x_1 \mid i \leq n\}$, de variables aleatorias con r estados posibles.
- Θ es el conjunto $\{\Theta_i \mid i \leq n\}$ y $\Theta_i = \{P(x_i = k \mid \pi_i = j) \mid k \in Q$ y j es una configuración de los padres de $x_i\}$ donde $P(x_i = k \mid \pi_i = j)$, denota la probabilidad de que el estado x_i sea k, dado que la configuración de padre es j.

Una red bayesiana representa en forma gráfica las dependencias e independencias entre variables aleatorias, en particular las dependencias condicionales.

4 Ruido

El ruido se agrega a la distribución de datos para conservar la privacidad, el objetivo del ruido es que la probabilidad de obtener algún resultado de los datos sea parecida a la que se podría obtener de otro conjunto de datos siendo distintos del original en un elemento. Con el software se implementa un mecanismo diferencialmente privado, agregando ruido controlado a las distribuciones aprendidas. El ruido se agrega con la distribución de Laplace que abordamos con más detalle a continuación.

4.1 Privacidad diferencial

La privacidad diferencial es una familia de técnicas que garantiza que la salida de un algoritmo es estadísticamente indistinguible en un par de bases de datos vecinas, esto quiere decir que el par de base de datos difieren solo en una tupla [6]. Así mismo, Privacidad diferencial consiste en recibir dos bases de datos que difieren exactamente en una fila y que con la ayuda de un algoritmo diferencialmente privado proporcionará resultados aleatorios que siguen distribuciones de probabilidad casi idénticas en ambas bases de datos [7]. Con ε se puede estimar la variación de la privacidad de algún individuo ya sea que su información este contenida en una base de datos de algún estudio. Vamos a considerar que D es la base de datos donde se podría encontrar dicha información del sujeto x, y D' será la base de datos en la cual se eliminó la fila que contiene los datos del sujeto, o se sustituyó por otra BD. La privacidad diferencial nos garantiza poder ocultar los datos de cada sujeto minimizando el daño y sin afectar la utilidad de los datos [8].

El concepto de la privacidad diferencial se formaliza en la siguiente definición:

- Sea ε un número positivo. Determina la proximidad de los resultados aleatorios en ambas bases de datos. Entre más alto el valor de ε implica que se conserve la privacidad.
- Sea A un mecanismo o algoritmo al azar teniendo un conjunto de datos como entrada.
- Sea D un conjunto de datos.
- Para cualquier D' que difiere de D en a lo más una tupla.
- Para cualquier salida S de A:

$$\mathrm{P}(\,\mathrm{A(D)} = \mathrm{S}\,) \leq e^{\varepsilon} \times \mathrm{P}(\,\mathrm{A(D')} = \mathrm{S}\,) \tag{2}$$

Cuando ε se acerca a 0, P(A(D) = S) = P(A(D') = S), esto quiere decir que la presencia o ausencia de un solo individuo en la entrada al algoritmo será indetectable cuando uno observa la salida.[6]

La privacidad diferencial requiere que la ganancia de conocimiento esté limitada por e^{ε}.

4.2 Ruido de Laplace

El mecanismo de Laplace utiliza la función de distribución de Laplace para introducir el ruido a los datos y cumpla la privacidad diferencial ε.

La distribución de Laplace es la distribución de la diferencia de dos variables aleatorias independientes con distribuciones exponenciales idénticas.

La distribución de Laplace con parámetro μ y escala b > 0 es la función de densidad.

$$Lap(z|\mu, \mathrm{b}) = \frac{1}{2b}\exp\,(-\frac{|z-\mu|}{b}) \tag{3}$$

La distribución de Laplace es una versión simétrica de la distribución exponencial. Tiene esperanza o promedio μ y varianza $\sigma^2 = 2b^2$. Se puede referir como Lap(b) (asumiendo que μ = 0) para indicar que hablamos de alguna distribución X ~ Lap(0, b). [8]

La distribución de la cual se muestrea un atributo dependiente se llama distribución condicionada.

Ahora, el software DataSynthesizer para preservar la privacidad diferencial introduce las distribuciones condicionadas con ruido:

$$Lap(\frac{4(d-k)}{n \cdot \varepsilon}) \tag{4}$$

Donde d es el número de atributos, k es número máximo de padres de un nodo de una red bayesiana y n es el número de tuplas en el conjunto de datos de entrada.

5 DataSynthesizer

DataSynthesizer (DS) es un sistema integral que toma un conjunto de datos privados como entrada y genera conjuntos de datos sintéticos, que simulan un conjunto de datos. El sistema está implementado en Python 3[14]. Su objetivo es facilitar las colaboraciones entre científicos de datos y propietarios de datos confidenciales. Aplica técnicas de Privacidad Diferencial para lograr una fuerte garantía de privacidad.

Captura la estructura de correlación subyacente entre los diferentes atributos mediante la construcción de una red bayesiana, Las cadenas no categóricas permiten a DS generar cadenas aleatorias durante la etapa de generación de datos. Esta característica permite a DataSynthesizer crear conjuntos de datos que se parecen a la muestra real al incluir datos de cadenas sintéticas como nombres artificiales e identificaciones.

DataSynthesizer puede operar en tres modos:

- Correlation mode: Construye una red bayesiana diferencialmente privada que captura la estructura de correlaciones entre atributos y, a continuación, extrae muestras de este modelo para construir el conjunto de datos resultante.
- Independent attribute mode: En este modo se obtiene un histograma para cada atributo, se agrega el ruido al histograma para conseguir la privacidad diferencial y se extraen muestras para cada atributo. Se usa cuando el modo de atributos correlacionados es demasiado caro computacionalmente o cuando no hay datos suficientes para derivar un modelo razonable
- Random mode: Simplemente genera valores aleatorios consistentes con el tipo para cada atributo, es para casos de datos extremadamente sensibles.

5.1 Módulos

A continuación, se presentan los tres módulos del sistema y su descripción.

5.1.1 DataDescriber: Elaborar un resumen de datos

El conjunto de datos de entrada es procesado primero por el módulo DataDescriber, guarda una descripción del conjunto de datos en un archivo JSON.

DataDescriber investiga los tipos de datos, las correlaciones y las distribuciones de los atributos en el conjunto de datos privados, y elabora un resumen de los datos, añadiendo ruido a las distribuciones para preservar la privacidad.

Primero el conjunto de datos de entrada es procesado por el módulo DataDescriber. Los dominios y las estimaciones de las distribuciones de los atributos se infieren y se guardan en un archivo de descripción del conjunto de datos.

DataDescriber requiere parámetros para tener un mejor ajuste de la descripción de datos, uno de estos parámetros es el umbral categórico, como cualquier umbral, puede ser difícil de establecerlo de forma que refleje las preferencias del usuario.

Otro parámetro importante es épsilon, un parámetro de privacidad diferencial. Significa que eliminar una fila del conjunto de datos de entrada no cambiará la probabilidad de obtener el mismo resultado más que una diferencia multiplicativa de exp(épsilon). Se aumenta el valor de épsilon para reducir los ruidos inyectados. Establecer épsilon=0 para desactivar la privacidad diferencial.

5.1.2 DataGenerator: Generar un conjunto de datos sintéticos a partir del resumen

DataGeneretor toma muestras de la distribución de frecuencias de los valores calculados con DataDescriber para obtener los datos sintéticos.

En este módulo se selecciona el modo de operar del software, ya sea Independent attribute mode (modo de atributo independiente), Correlation mode (modo correlacionado) o random mode (modo aleatorio); Cuando se invoca el modo aleatorio DataGenerator genera valores aleatorios de tipo coherente para cada atributo. En el modo de atributo independiente extrae muestras de gráficos de barras o histogramas mediante muestreo uniforme. Y cuando se hace el llamado al modo correlacionado se muestrean los valores de los atributos conservando la matriz de correlaciones de los datos generados a partir de la red bayesiana.

5.1.3 ModelInspector: Inspección y comparación de conjuntos y resúmenes de datos

ModelInspector muestra una descripción intuitiva del archivo description.json de datos calculado por DataDescriber, lo que permite al propietario de los datos evaluar la precisión del proceso de generación y ajustar los parámetros, si lo desea.

Proporciona varias funciones integradas para inspeccionar la similitud entre el conjunto de datos privados de entrada y el conjunto de datos sintéticos de salida. El propietario de los datos puede comprobar rápidamente si las tuplas del conjunto de datos sintéticos son detectables inspeccionando y comparando las 5 primeras y las 5 últimas tuplas de ambos conjuntos de datos.

5.2 Algoritmos

La distribución condicional es construida de acuerdo con el algoritmo 1 de [8]. En este algoritmo se encarga de agregar el ruido de Laplace a los atributos y el algoritmo 2 de [6] describe el proceso de la generación de datos.

Algoritmo 1: Ruido Condicional (D, N, k): regresa P *.

1.Inicializar variable para guardar la distribución condicional de salida, P* = 0;
2.para i = k +1 hasta d hacer: #i = k+1 hasta número de atributos.
3. materializar la distribución conjunta Pr[X_i, $\prod_i$]; #Se genera el espacio para la distribución condicional del hijo X_i.
4. se genera la privacidad diferencial Pr*[X_i, $\prod_i$] agregando el ruido de Laplace;
5. establecer los valores negativos de la privacidad diferencial Pr*[X_i, $\prod_i$] a 0 y normalizar;
6. obtener la distribución condicional con ruido Pr*[$X_i \mid \prod_i$] de la distribución en Pr*[X_i, $\prod_i$]; agregarlo a P*;
7.para i = 1 hasta k hacer:
8. obtener la distribución condicional con ruido Pr*[$X_i \mid \prod_i$] de Pr*[X_{k+1}, $\prod_{k+1}$]; se agrega a P*;
9.regresa P *;

Algoritmo 2: DataGenerator (n, M, S, A_u., *s)*

Se requiere: número de tuplas n a generar, modo M, descripción del conjunto de datos S, atributos uniformes A_u, semilla s.

```
1 Establecer semilla = s para el generador de números
pseudoaleatorios.
2 si M es modo de atributos independientes (independent attribute
  mode) entonces:
3   Leer todos los atributos A de S.
4     para X ∈ A hacer #Se hace recorrido por los atributos
5       si X ∈ A_u entonces:
6         Leer el dominio de X a partir de S.
7         Muestrea n valores uniformemente de su dominio.
8       sino
9         Lee la distribución de X a partir de S.
10        Muestrea n valores de su distribución.
11      fin si
12     fin para
13  sino si M es modo de atributos correlacionados (correlated
    attribute mode) entonces:
14    Leer la red bayesiana N a partir de S.
15    Muestra atributo raíz de una distribución No condicional.
16    Muestra atributos restantes de distribución condicional.
17  fin si
18  regresa el Conjunto de datos muestreados;
```

6 Experimentación

6.1 Recuperación de datos

Se recupera la base de datos Adult Data Set de UCI Machine Learning Repository [http://archive.ics.uci.edu/ml] [10][11]. Esta base de datos contiene información de los ingresos de adultos basándose en los datos del censo. También conocido como conjunto de datos "Census Income".

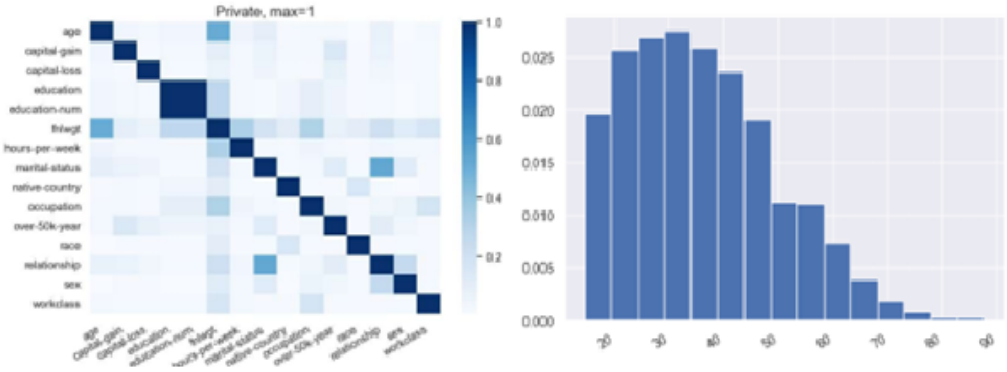


Figura 1. Matriz de correlación de atributos de datos originales: Adult Data Set y Distribución de probabilidad de atributo Age: Adult Data Set [10].

6.2 Resultados

Se generan los datos sintéticos con la biblioteca DataSynthesizer de Python, se puede encontrar el código fuente y aplicaciones en ejemplos en [9][12][13].
A continuación, se presentan algunas observaciones. Para el número máximo de padres para cada nodo, se observa que un mejor ajuste es cuando cada nodo tiene máximo dos padres.

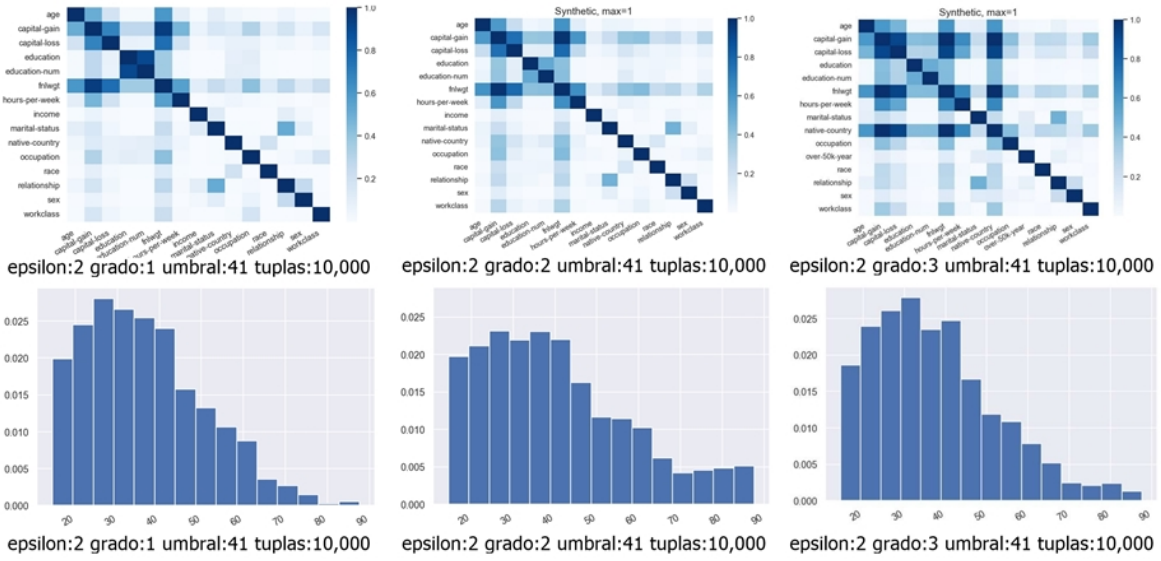


Figura 2. Comparación de resultados para el parámetro del grado de la red bayesiana.

También, se observa que cuando se cambia el grado de la red bayesiana, no hay cambios sustanciales ni en la matriz de correlación ni en el histograma.
En la matriz de correlación para los datos sintéticos, no se observan grandes diferencias en los datos sintéticos, a pesar de que se aumenta el número de datos sintéticos. En cambio, en el histograma, al aumentar el número de datos sintéticos a generar se parece más su distribución a la original.

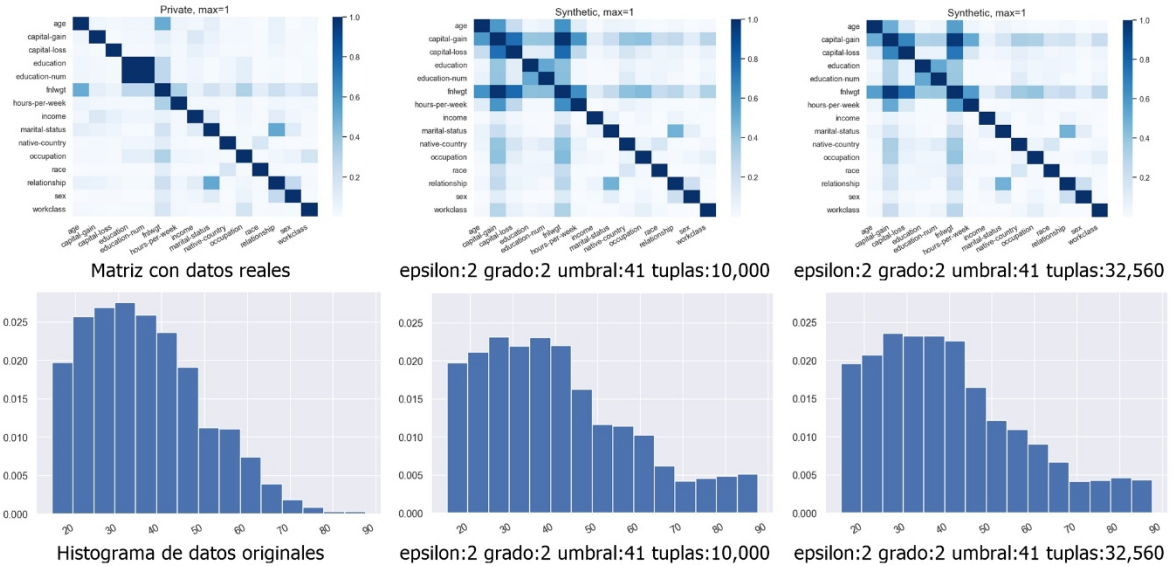


Figura 3. Comparación cuando el número de datos sintéticos generados es diferente.

Después de obtener resultados con diferente cantidad de datos generados, se concluye que se tiene un mejor comportamiento de los datos sintéticos entre más alto el número de tuplas a obtener.
De acuerdo con la experimentación se observa que un mejor comportamiento en las variables es cuando el umbral es cercano al dominio del atributo categórico con más valores diferentes. En este caso es el atributo 'native-country' igual a 40.

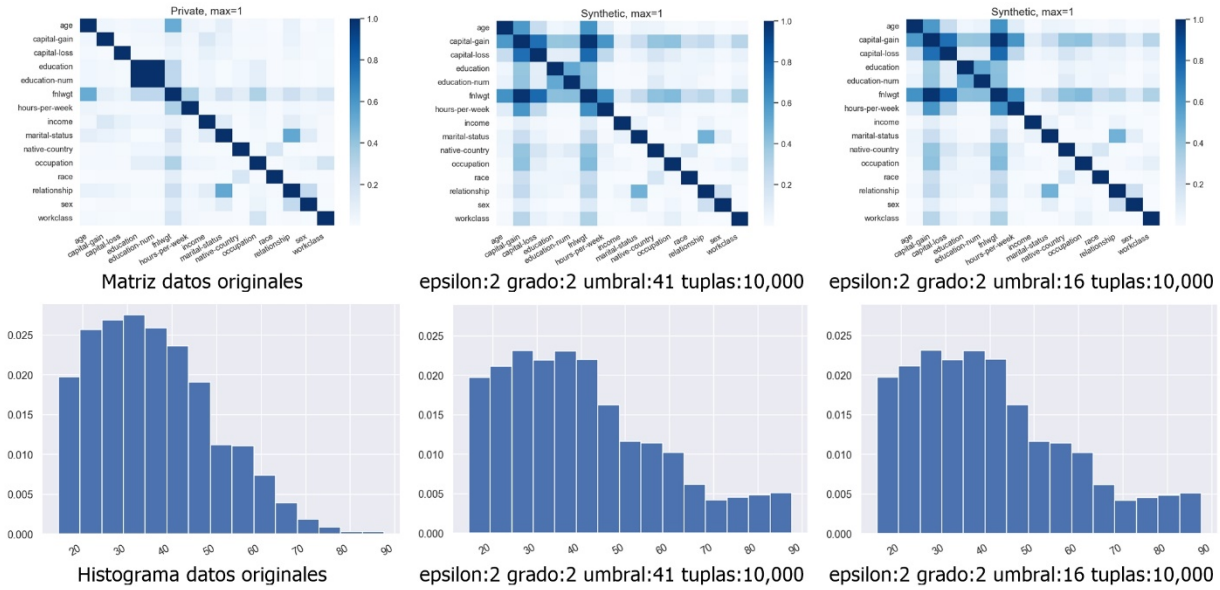


Figura 4. Comparación cuando el umbral es diferente.

Con antelación se menciona la utilidad del valor épsilon, y para el software DataSynthesizer se debe tener en cuenta qué tanto ruido se desea ingresar a los datos, ya que con un ruido mayor la distribución entre datos originales y sintéticos NO tendrá una diferencia significativa, mientras que la matriz de correlación queda determinada por el grado de la red bayesiana.

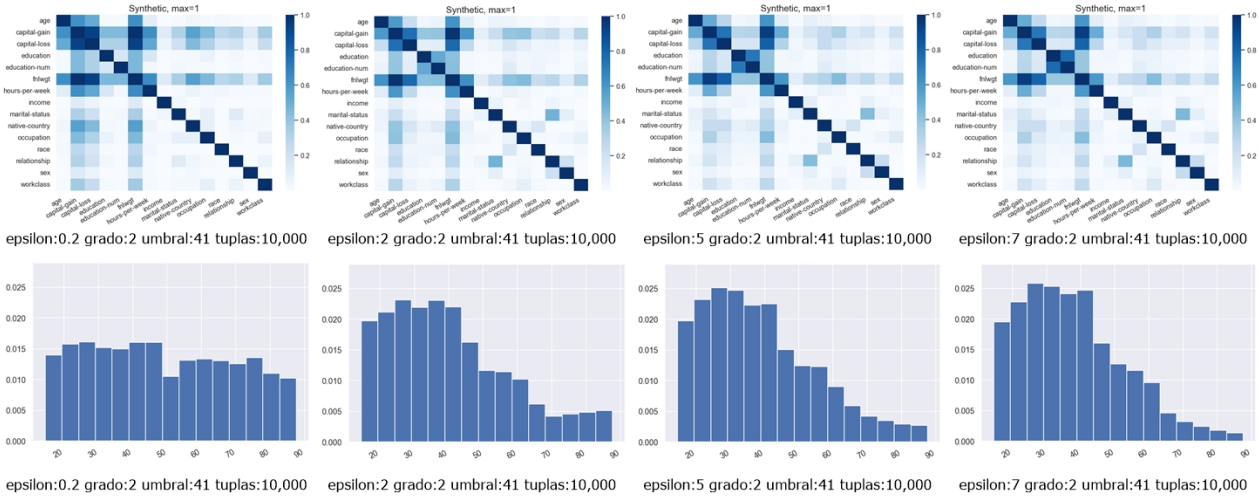


Figura 5. Comparación entre diferentes valores para épsilon.

7 Conclusiones

Los datos sintéticos nos ayudan a entrenar modelos de inteligencia artificial cuando los datos son insuficientes o para no comprometer información confidencial, resultando en datos con utilidad para diferentes proyectos. Entre una variedad de software para generar datos sintéticos se tiene DataSynthesizer, un software generador de datos sintéticos, utilizando ruido laplaciano y redes bayesianas. Se muestra que para la base de datos Adult Data Set [10], cuando el ruido laplaciano es mayor, se observa que se conserva la distribución de los atributos, además de la confidencialidad de estos. Mientras que cuando se cambia el grado de la red bayesiana se modifica la matriz de correlación.

Referencias

[1] Yan I. Kuchin, Ravil I. Mukhamediev & Kirill O. Yakunin | Duc Pham (Reviewing editor) (2020) One method of generating synthetic data to assess the upper limit of machine learning algorithms performance, Cogent Engineering, 7:1, DOI: 10.1080/23311916.2020.1718821.

[2] Debbie Rankin, Maurice Mulvenna, Michaela Black, Raymond Bond, Jonathan Wallace, Gorka Epelde (2020). Reliability of Supervised Machine Learning Using Synthetic Data in Health Care: Model to Preserve Privacy for Data Sharing. JMIR Med Inform; 8(7) :e18910. DOI: 10.2196/18910.

[3] Fernando Fuentes (2022). Datos sintéticos, un recurso vital para la Inteligencia Artificial. [Online]. Available: https://www.arsys.es/blog/datos-sinteticos. [Accessed: 27/Septiembre/2022].

[4] Dankar, F.K.; Ibrahim, M. Fake It Till You Make It: Guidelines for Effective Synthetic Data Generation. Appl. Sci. 2021, 11, 2158. DOI: 10.3390/app11052158.

[5] Ronald E. Walpole, Raymond H. Myers, S. L. M. Y. K. Y. (2012): Probabilidad y estadística para ingeniería y ciencias, novena edición Pearson Educación, México, 2012, Col. Industrial Atoto, Naucalpan de Juárez, Estado de México.

[6] Haoyue Ping, Julia Stoyanovich, and Bill Howe (2017). DataSynthesizer: Privacy-Preserving Synthetic Datasets. In Proceedings of SSDBM ’17, Chicago, IL, USA, June 27-29, 2017, 5 pages. DOI: 10.1145/3085504.3091117.

[7] Fida K. Dankar, Khaled El Emam (2013). Practicing Differential Privacy in Health Care: A Review. Trans. Data Privacy 6, 1 (April 2013), 35–67. DOI: 10.5555/2612156.2612159.

[8] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. 2017. PrivBayes: Private Data Release via Bayesian Networks. ACM Trans. Database Syst. 42, 4, Article 25 (December 2017). https://doi.org/10.1145/3134428.

[9] Ping, H., & Yang, K. (2020, 11 junio). DataSynthesizer. GitHub. [Online]. Available: https://github.com/DataResponsibly/DataSynthesizer. [Accessed: 14/Enero/2023].

[10] Kohavi, Ronny. Becker Barry(1996). Adult Data Set. UCI Machine Learning Repository. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/adult. [Accessed: 23/Noviembre/2022].

[11] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository. [Online]. Available: http://archive.ics.uci.edu/ml. Irvine, CA: University of California, School of Information and Computer Science. [Accessed: 23/Noviembre/2022].

[12] Haoyue, Ping. DataSynthesizer Usage (correlated attribute mode). correlated_attribute_mode.ipynb. [Online]. Available: https://github.com/DataResponsibly/DataSynthesizer/blob/master/notebooks/DataSynthesizer__correlated_attribute_mode.ipynb. [Accessed: 15/Octubre/2022].

[13] Bohorquez, Nicolas. synthetic-data(DataSynthesizer). [Online]. Available: synthetic-data.ipynb. https://github.com/nickmancol/synthetic-data/blob/3184d35285147e005125b72c6cc65423e174151a/synthetic-data.ipynb. [Accessed: 23/11/2022].

[14] Python3. [Online]. Available: https://www.python.org/. [Accessed: 3/08/2022].

Desarrollo de una aplicación móvil con realidad aumentada como apoyo a la enseñanza de la biología

Mobile application development with augmented reality to support the teaching of biology

Ana L. Ballinas Hernández[1], Victor H. Martínez Zepeda[1], M. Claudia Denicia Carral[1], Maricruz Rangel Galván[2]

[1]Complejo Regional Centro, Benemérita Universidad Autónoma de Puebla. Calle 2 Sur S/N, Ciudad Modelo. C.P. 75012. San José Chiapa, Puebla, México.

[2]Facultad de Ciencias Químicas, Benemérita Universidad Autónoma de Puebla. Av. San Claudio, Ciudad Universitaria. C.P. 72570. Puebla, México.

analuisa.ballinas@correo.buap.mx, victor.martinezz@alumno.buap.mx, claudia.denicia@correo.buap.mx, maricruz.rangelgalvan@viep.com.mx.

Abstract

Teaching methods have drastically transformed in recent years and new innovative technologies have been adopted to enhance the student experience. In this work, an application is developed that uses augmented reality to increase motivation in the learning process of Biology at a basic level. The main result obtained is a mobile application that superimposes virtual elements, such as 3D models, virtual assistant with voice, text, and multimedia videos, with the physical world, which allows users to visualize and interact realistically with didactic content on biology topics. As a contribution of the work, a methodological proposal is presented to develop support applications for teaching through augmented reality that achieves significant learning in students.

Resumen

Los métodos de enseñanza se han transformado drásticamente en los últimos años y por ello se han adoptado nuevas tecnologías innovadoras que mejoran la experiencia de los estudiantes. En este trabajo se desarrolla una aplicación que usa realidad aumentada para incrementar la motivación en el proceso de aprendizaje de la materia de Biología a nivel básico. El principal resultado obtenido es una aplicación móvil que superpone elementos virtuales, como modelos 3D, asistente virtual con voz, texto y videos multimedia, con el mundo físico lo cual permite a los usuarios visualizar e interactuar de forma realista con contenidos didácticos de temas de biología. Como aportación del trabajo se presenta una propuesta metodológica para desarrollar aplicaciones de apoyo a la enseñanza mediante realidad aumentada que logre un aprendizaje significativo en los estudiantes.

Keywords and phrases: Realidad Aumentada, Innovación Tecnológica en la Educación, Enseñanza de la Biología, Aplicaciones Móviles.

---

1 Introducción

La educación se ha transformado drásticamente durante los últimos años y por ello se han adoptado nuevas técnicas para el autoaprendizaje. Los métodos tradicionales de enseñanza hacen poco uso de la tecnología de realidad aumentada (RA) desaprovechando todos los beneficios que esta brinda.

La RA combina en tiempo real el mundo físico con elementos virtuales por medio de la cámara de algún dispositivo inteligente al reconocer patrones de una imagen u otro elemento. Esta tecnología puede utilizarse para generar diferentes experiencias y para modificar la interacción entre los usuarios y las aplicaciones. El uso de estas tecnologías tiene grandes beneficios en el ámbito de la educación, tales como: el aumento del interés de los alumnos, la mejora en la comprensión de los contenidos y el aumento del nivel de aprendizaje en los temas abordados.

Además, la RA estimula el estilo de aprendizaje visual debido a que se usan representaciones gráficas, como imágenes y modelos tridimensionales, que pueden ser relacionados con conceptos e ideas y con la capacidad de abstracción de las personas, logrando un aprendizaje más eficiente [1]. El uso del aprendizaje visual en aplicaciones de RA propicia que los alumnos puedan organizar sus ideas, priorizar información, incrementar la comprensión y al mismo tiempo estimular el pensamiento creativo.

En este trabajo se desarrolla una aplicación móvil para interactuar con contenidos didácticos de la materia de biología a nivel básico mediante realidad aumentada. La aplicación incluye un asistente virtual con voz para que los usuarios interactúen de forma realista e intuitiva, así como elementos virtuales como modelos 3D y videos didácticos. Para superponer los elementos virtuales en el mundo físico, se escanean marcadores, correspondientes a imágenes del libro de biología de primer año de secundaria, usando la cámara del dispositivo móvil. Como principal aportación, se presenta una propuesta metodológica para desarrollar aplicaciones de apoyo a la enseñanza mediante el uso de realidad aumentada estimulando el estilo de aprendizaje visual.

2 Trabajo relacionado

El uso de aplicaciones de realidad aumentada empleadas para la educación ha crecido significativamente en los últimos años. La RA se considera como un recurso de apoyo a la enseñanza donde las personas pueden desarrollar su creatividad y capacidades de autoestudio [2].

Se ha desarrollado una aplicación para dispositivos móviles que permite concientizar a niños en etapas tempranas sobre la conservación del medio ambiente. Para ello, se empleó la tecnología de RA que brinda un aspecto interactivo y apoya en el proceso de enseñanza-aprendizaje en la asignatura de ciencias naturales [3]. Para el funcionamiento de esa aplicación se utilizaron varias láminas relacionadas con sitios naturales de las regiones mostrando información, animaciones y objetos 3D relacionados con los temas. Asimismo, se desarrolló una guía didáctica ordenada de los temas abordados en los libros de Ciencias Sociales y Ciencias Naturales, con el objetivo de que los alumnos tengan una aplicación de fácil entendimiento y de fácil uso [4]. Por cada tema que se presenta en la guía, se muestra un modelo 3D con información y sonido narrativo.

Las aplicaciones de realidad aumentada tienen un gran potencial ya que las tecnologías crecen de forma exponencial, causando un impacto positivo en la sociedad. En el trabajo presentado por Castellano y Santacruz se desarrolló una aplicación basada en RA que contiene un juego educativo

de preguntas y respuestas para niños de entre 6 y 9 años [5]. Al final de las pruebas se realizó un *test* para comparar los resultados e identificar mejoras en el aprendizaje de cada niño.

La posibilidad de interacción con aplicaciones es más enriquecedora para los usuarios por el hecho de que pueden manipular objetos en todo momento [6]. Un elemento importante en la educación es el nivel de motivación del estudiante que estimula y propicia una mejora en el aprendizaje. En ese trabajo se presentan tres fases, donde los alumnos llegan a mostrar motivación e interés, curiosidad y un enfoque activo. Con el uso de esa aplicación se muestra que para captar el interés de un niño se deben despertar sus emociones y de esa forma tener mejor comprensión de los temas.

Aponte-Zurita construyó una aplicación relacionada con los dinosaurios mediante realidad virtual (RV). En comparación con la RA, que solo necesita un teléfono móvil e imágenes como marcadores, la RV requiere varios dispositivos como gafas de RV, *Gear* VR control y un Smartphone [7]. El desarrollo de esa aplicación se hizo aplicando la metodología XP, la cual garantiza el buen desarrollo de software ágil y la evaluación de la usabilidad de la aplicación. Las pruebas fueron realizadas con una población de 30 alumnos arrojando valores superiores a la evaluación diagnóstica.

En el artículo presentado por Sousa *et al*., se muestra como la realidad virtual ha sido empleada en varios campos profesionales y educativos resaltando el área de medicina, educación, electrónica y computación [8]. A pesar de que la mayoría de los resultados fueron positivos, se deben considerar algunas restricciones como: el costo de los dispositivos a utilizar, la precisión de los dispositivos y tener un entorno controlado entre estudiantes y docentes para poder ejecutar las aplicaciones.

En el trabajo presentado por Juca y colaboradores se presenta un proyecto de rutas inmersivas dirigido a estudiantes de turismo para comprender más fácilmente temas de sus asignaturas [9]. Después de usar la aplicación de RV, se comprobó que los estudiantes fueron más participativos y que tuvieron mayor interés en los temas. En general, existen pocas aplicaciones que abordan temas de biología mediante el uso de nuevas tecnologías. El presente trabajo se enfoca al desarrollo de una aplicación que estimule el interés por estudiar la materia de biología aprovechando los beneficios de la RA en un contexto de innovación educativa y tecnológica.

3 Desarrollo de la aplicación

En este trabajo se desarrolla una aplicación móvil para dispositivos con sistema operativo Android como apoyo en el aprendizaje de la asignatura de biología a nivel secundaria mediante el uso de realidad aumentada. Para el desarrollo de la aplicación se siguen las fases de la propuesta metodológica mostradas en la Figura 1. A continuación se describe cada fase y como fue realizada.

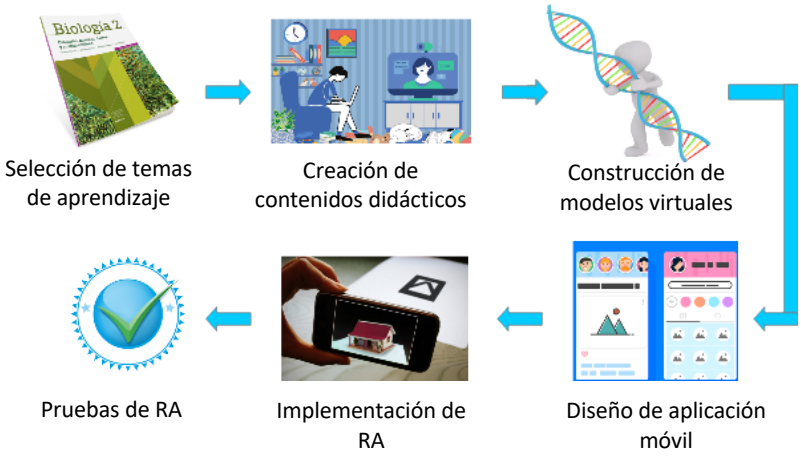


Figura 1. Fases metodológicas para el desarrollo de la aplicación.

3.1 Selección de temas de aprendizaje

En esta fase se delimitan los bloques del curso de biología que aborda la aplicación siguiendo el plan de estudios del nivel secundaria. La selección de temas se realizó mediante la aplicación de una encuesta en línea a una población de 234 estudiantes de 6 escuelas públicas del estado de Puebla a nivel secundaria. La Figura 2 muestra el resultado de los temas con más dificultad para los encuestados. Por lo tanto, los tres temas que se abordan en la aplicación mediante el uso de realidad aumentada son: la célula (unidad estructural y funcional de los seres vivos), el interior de la célula y las Interacciones.

Figura 2. Encuesta para selección de temas de aprendizaje.

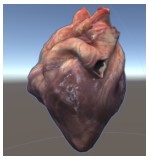
3.2 Creación de contenidos didácticos

En esta fase se exploran los temas seleccionados y la forma en que mejor se adaptan a la realidad aumentada. Por ello, se construyen contenidos de los temas didácticos que permitan a los usuarios visualizar e interactuar de forma realista con la aplicación. El objetivo de esta fase consiste en aplicar el enfoque constructivista para que la aplicación sirva como herramienta para que los alumnos desarrollen su capacidad de aprender a construir sus propios conocimientos y a su vez puedan

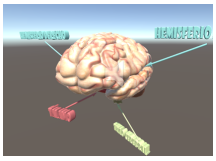
relacionarlos con su entorno combinado con elementos virtuales [10]. Se considera un proceso de enseñanza centrado en el estudiante para favorecer su conocimiento, habilidades y competencias mediante la aplicación de estrategias lúdicas que despiertan el interés de los temas de estudio aplicando un estilo de aprendizaje visual. Para ello, se construyen diversos elementos virtuales para generar contenidos didácticos visuales que son mostrados mediante realidad aumentada para motivar el interés de los alumnos y lograr un aprendizaje significativo.

3.3 Construcción de modelos virtuales

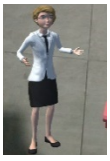
Para la construcción de contenidos didácticos de los temas seleccionados se incluye el uso de: texto, imágenes, modelos 3D de los temas abordados, animaciones, un asistente virtual, videos didácticos de enseñanza y voz. Los modelos tridimensionales son construidos y renderizados usando el software de modelado Cinema 4D. El asistente virtual es construido usando el software fuse para la creación de personajes. Los videos multimedia contienen imágenes y audio para describir los temas correspondientes. La Figura 3 muestra algunos ejemplos de los modelos 3D, videos e imágenes construidos para que sean usados por la aplicación de realidad aumentada.



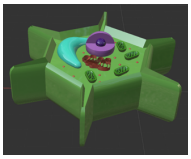
a) Modelo 3D de corazón



b) Modelo 3D del cerebro humano

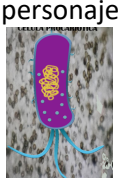


c) Modelo 3D del personaje



d) Modelo 3D de célula vegetal

e) Video de neurona del cerebro

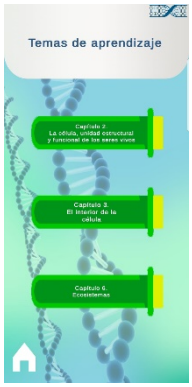


f) Imágenes

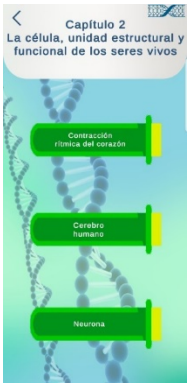
Figura 3. Elementos virtuales para la creación de contenidos didácticos.

3.4 Diseño de la aplicación móvil

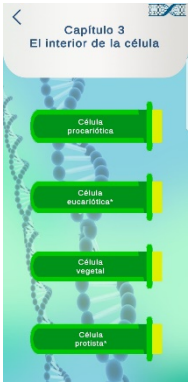
a) Menú principal



b) Temas de aprendizaje



c) Primer tema de aprendizaje



d) Segundo tema de aprendizaje

Figura 4. Interfaces de usuario de la aplicación móvil.

Se realizó el diseño de la aplicación móvil considerando los temas de la materia de biología seleccionados. El diseño está inspirado en los libros de biología de nivel secundaria. Para el diseño de las interfaces móviles se empleó el motor multiplataforma Unity usado para el desarrollo de videojuegos. La Figura 4 muestra las interfaces de usuario de la aplicación móvil la cual puede ser instalada en dispositivos con sistema operativo Android.

3.5 Implementación de RA

Esta fase consiste en la selección de los marcadores que al ser escaneados muestran elementos virtuales mediante realidad aumentada. Estos marcadores deben contener un patrón irregular para que sean fáciles de reconocer por la cámara del dispositivo. Cada marcador es obtenido del libro de textos de biología del primer año de secundaria. La Figura 5 muestra algunos marcadores empleados para visualizar la realidad aumentada.

Figura 5. Ejemplos de marcadores empleados para visualizar la RA.

Para la implementación de la aplicación se utilizó el software Unity, el lenguaje de programación basado en scripts C# y la plataforma Vuforia. Esta plataforma es un kit de desarrollo que permite a desarrolladores integrar fácilmente tecnologías de realidad aumentada y mixta a las aplicaciones usando diferentes dispositivos [11].

En general, la selección de la herramienta a utilizar para desarrollar aplicaciones de realidad aumentada depende de los requerimientos y de la compatibilidad con los dispositivos móviles empleados. La Tabla 1 muestra una descripción de algunas herramientas y bibliotecas usadas para construir aplicaciones de realidad aumentada en diferentes plataformas de desarrollo tales como el motor de juegos Unity, Unreal, Android Studio y Visual Studio bajo dispositivos con Android o iOS. Algunas de estas herramientas son de uso libre y algunas otras de uso comercial.

3.6 Pruebas de RA

Se han realizado pruebas de funcionalidad en cinco dispositivos con sistema operativo Android 4.4 o superior y se ha validado que la realidad aumentada sea visualizada adecuadamente. Para poder usar la aplicación se requiere la autorización de permisos para usar la cámara, tener disponible 500 MB en el dispositivo y contar con giroscopio. Todos los marcadores fueron probados para verificar que muestran los elementos virtuales correctamente al ser escaneados con la cámara. Los errores de enfoque de cámara y de elementos virtuales fueron corregidos y la aplicación funciona correctamente. Los resultados obtenidos son presentados y discutidos en la siguiente sección.

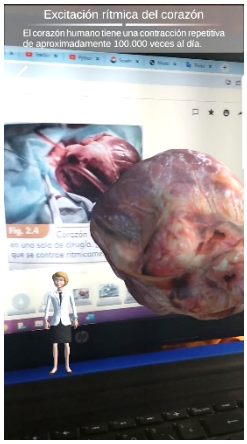
Tabla 1. Herramientas empleadas para el desarrollo de aplicaciones con realidad aumentada.

Biblioteca	Descripción	Plataformas	Licencia
Vuforia Engine	Kit de desarrollo de aplicaciones básicas y avanzadas de Realidad Aumentada y Realidad Mixta.	Android, iOS, Unity, Android Studio, Visual Studio, Xcode	Libre y comercial
AR Foundation	Biblioteca que permite crear aplicaciones en Unity tales como: seguimiento de planos, de gestos, de rostros, de imágenes y generación de nubes de puntos.	Unity, Android, iOS	Libre
AR Core	Plataforma de Google para crear experiencias de RA permitiendo que el teléfono detecte e interactúe con el entorno.	Android, iOS, Android Studio, Unreal	Libre
ARkit	Framework para crear experiencias físicas y aumentadas solo para dispositivos Apple.	iOS	Libre y comercial
ARToolKit	Kit de herramientas de software de RA con código fuente abierto que implica el acceso libre.	Android, iOS, Linux, Unity	Libre
WikiTude	Biblioteca para crear apps de RA, reconstrucción de lugares en mapas virtuales, tecnologías de seguimiento y geolocalización.	Google Glass, Epson Moverio, Vuzix, PhoneGap, Titanium	Comercial
LayAR	Plataforma para visualizar escenarios superponiendo diferentes capas de RA.	Android, iOS, BlackBerry, Android Studio	Comercial
Kudan	Biblioteca que permite mapear modelos multi poligonales en la realidad física e importarlos a modelos 3D.	Android, iOS, Unity	Comercial

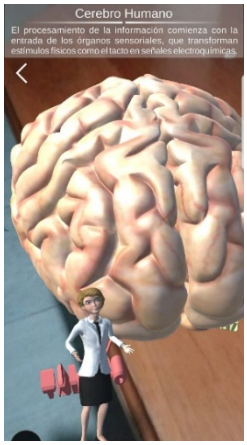
4 Resultados

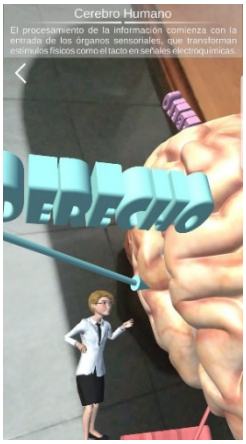
En esta sección se presentan los resultados obtenidos de la implementación de la propuesta metodológica mostrada en la sección anterior. El principal resultado obtenido es una aplicación móvil para dispositivos con Android de apoyo para la materia de biología, así como los elementos virtuales que son mostrados mediante realidad aumentada.

Al ejecutar la aplicación se deben seleccionar los temas de aprendizaje mostrados en la Figura 4. Al seleccionar algún tema se habilita la cámara y un asistente virtual con voz guía a los usuarios en el uso de la aplicación y describe los contenidos didácticos. Para visualizar la realidad aumentada es necesario escanear el marcador relacionado con el tema seleccionado que corresponde a una figura del libro de biología de primer grado de secundaria. Una vez que la cámara reconozca el patrón del marcador, se muestran los elementos virtuales correspondientes a los contenidos didácticos que son superpuestos al entorno real. La Figura 6 muestra la funcionalidad de la RA para el segundo capítulo de los temas de aprendizaje. En esta figura se incluye: un modelo 3D de la contracción del corazón, una animación del modelo de cerebro humano y un video didáctico que describe el tema de la neurona.

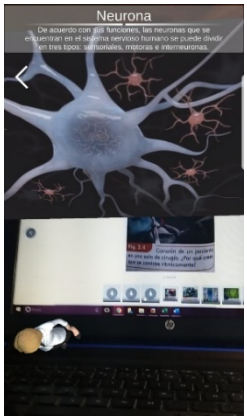


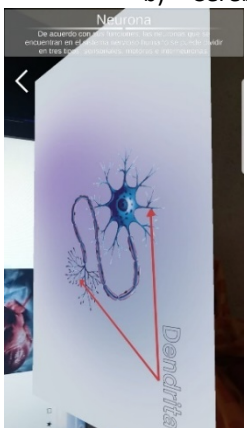
a) Contracción rítmica del corazón

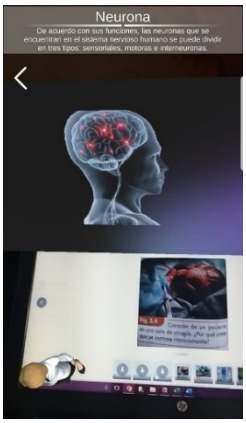




b) Cerebro humano





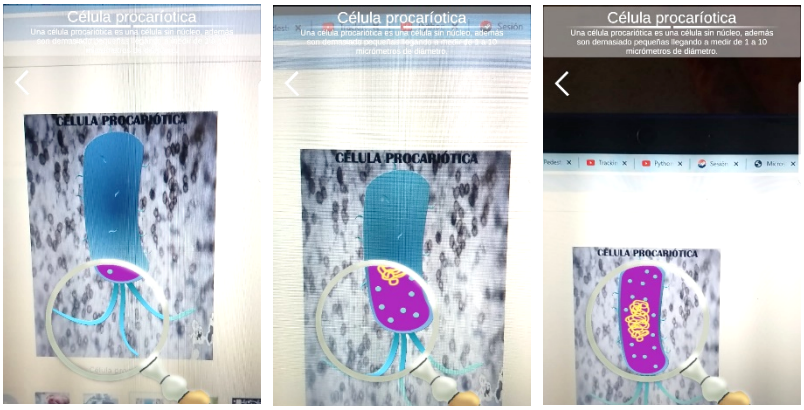


c) Video de neurona mostrado con RA

Figura 6. Unidad de aprendizaje: La célula, unidad estructural y funcional de los seres vivos. El asistente virtual describe los contenidos temáticos y se muestran los elementos virtuales.

La Figura 7 muestra el funcionamiento para el tercer capítulo denominado: El interior de la célula. Los temas incluidos son: la célula procariota y la célula vegetal. En la primera, se muestra un efecto de rayos X para escanear como se encuentra constituida una célula en su interior comprendiendo la información con el asistente de voz. En el segundo tema, se muestra un modelo 3D de la célula vegetal y el efecto *chroma key* para interponer el fondo de la hoja con un video didáctico que incluye aspectos teóricos del tema simultáneamente.

Como se puede observar, la aplicación de RA funciona adecuadamente bajos las restricciones establecidas. Esta aplicación sirve de apoyo a la enseñanza de la biología a nivel básico mediante un enfoque constructivista estimulando el estilo de aprendizaje visual de los usuarios.



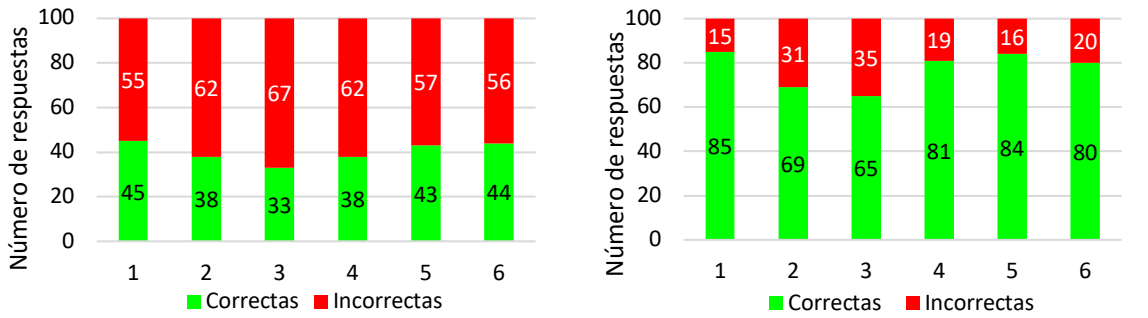
a) Célula procariota escaneada con efecto rayos X

b) Célula vegetal con efecto *chroma key*

Figura 7. Unidad de aprendizaje: El interior de la célula.

Para evaluar el desempeño de la aplicación se realizaron algunas pruebas con una muestra de 100 estudiantes de primer grado de secundaria de la Escuela Secundaria General “Justo Sierra” ubicada en Huamantla, Tlaxcala. Para ello, se construye un instrumento de evaluación que corresponde a un cuestionario de opción múltiple de seis preguntas obtenidas del libro de primer grado de secundaria para medir el nivel de conocimientos alcanzado. Este instrumento se aplicó en dos etapas: se evaluó el nivel de aprendizaje logrado mediante el uso de estrategias y métodos tradicionales de enseñanza de la biología usando el libro de textos; en la segunda etapa, los alumnos probaron la aplicación de realidad aumentada para interactuar con los temas didácticos y posteriormente se volvió a aplicar el instrumento de evaluación. La Figura 8 muestra una comparación de la cantidad de respuestas del cuestionario correctas e incorrectas obtenidas con y sin el uso de la aplicación.

Como se puede observar, el número de respuestas correctas aumenta después de que los estudiantes usaron la aplicación de realidad aumentada con respecto a técnicas de aprendizaje tradicional. Los resultados de la evaluación del uso de la aplicación revelan su utilidad debido a que se muestra una mejora en el proceso de aprendizaje de los estudiantes, a partir de experimentos realizados.



a) Resultados con estrategias didácticas tradicionales

b) Resultados después del uso de la aplicación

Figura 8. Comparación de los resultados obtenidos con un instrumento de evaluación antes y después de utilizar la aplicación.

5 Conclusiones

En este trabajo se desarrolló una aplicación móvil de apoyo a la enseñanza de la biología a nivel básico basada en realidad aumentada. Este tipo de innovaciones tecnológicas han revolucionado la educación debido a que son consideradas como grandes avances que mejoran la experiencia de los estudiantes en el proceso de aprendizaje.

Los resultados obtenidos en este trabajo son la construcción de modelos tridimensionales para describir contenidos didácticos de temas de la materia de biología a nivel básico que son superpuestos en mundo físico. Otro resultado es una aplicación móvil que permite que los usuarios visualicen e interactúen con los temas didácticos de forma realista y se logre un aprendizaje significativo aplicando un estilo visual. Además, se presenta una descripción de las principales herramientas y bibliotecas empleadas para implementar la realidad aumentada en aplicaciones de diversas áreas.

La principal aportación de este trabajo es la generación de una propuesta metodológica para la construcción de aplicaciones en el ámbito de educación mediante realidad aumentada que estimule el interés de los estudiantes. Esta aplicación es la base para la construcción de aplicaciones de enseñanza en temas más avanzados. Como trabajo a futuro derivado de este trabajo se propone la realización de pruebas de funcionalidad con una muestra más amplia de estudiantes de primer grado de secundaria y con un instrumento más completo que permita evaluar de forma más precisa el nivel de aprendizaje logrado como resultado del uso de la aplicación de realidad aumentada. Además, la implementación de un juego educativo de preguntas y respuestas con realidad aumentada para medir el desempeño de los estudiantes podría ser útil para evaluar el nivel de mejora del aprendizaje de la asignatura después de usar la aplicación.

Referencias

[1] C. J. H. Cardona, Aprendizaje visual, Revista Reforma Siglo XXI 29.112, 2022, 27-30.

[2] M. D. G. Zamar y E. A. Segura, La realidad aumentada como recurso creativo en la educación: Una revisión global. Creatividad y sociedad: revista de la Asociación para la Creatividad, 32, 2020, pp. 164-190.

[3] L. E. Muñoz Arracera y R. Montenegro Santos, Uso de la Realidad Aumentada en la enseñanza-aprendizaje de ciencias naturales. Memorias De Congresos UTP, 2017, pp. 96-101.

[4] J. G. Bautista Bonilla, Guía Didáctica aplicando Realidad Aumentada para alumnos de 4to año de Educación Básica, para las áreas de Ciencias Sociales y Ciencias Naturales en la Escuela Fiscal Mixta "Dr. Carlos Cadena N.". Marzo 23, 2022, de Bachelor's thesis, Quito: Universidad Central del Ecuador, 2018.

[5] T. Castellano Brasero y L. P. Santacruz Valencia. EnseñAPP: Aplicación Educativa de Realidad Aumentada para el primer ciclo de educación primaria. Revista Iberoamericana De Tecnología En Educación y Educación en Tecnología, 2018, pp. 7-14.

[6] R. Cozar Gutiérrez, y J. M. Sáez-López, Realidad aumentada, proyectos en el aula de primaria: experiencias y casos en Ciencias Sociales. EDMETIC, Revista de Educación Mediática y TIC, 6(1), 2017, pp. 165-180.

[7] W. R. Aponte Zurita, Aplicación de Realidad Virtual educativa sobre dinosaurios para niños de educación básica, Bachelor's thesis, Pontificia Universidad Católica del Ecuador, 2021.

[8] R. Sousa Ferreira, R. A. Campanari Xavier y A. S. Rodrigues Ancioto, La realidad virtual como herramienta para la educación básica y profesional, Revista Científica General José María Córdova, 2021.

[9] F. Juca Maldonado, J. Lalangui Ramírez, M. I. Bastidas Andrade, Rutas inmersivas de realidad virtual como alternativa tecnológica en el proceso educativo, Revista Metropolitana de Ciencias Aplicadas 3(1), 2020, pp. 48-56.

[10] C. J. M. Sócola, Estrategias de enseñanza basada en enfoque constructivista y evaluación de aprendizajes en Instituciones Educativas, Castilla-Piura, Revista de Educación 3(7), 2021, pp. 12-25.

[11] U. Technologies, (s. f.). Vuforia - Unity Manual. [Online]. Available: https://docs.unity3d.com/es/2018.4/Manual/vuforia-sdk-overview.html [Accessed: 27/Junio/2023].

Sistema de simulación del algoritmo perceptrón para redes neuronales en Python

Perceptrón algorithm simulation system for neural networks in Python

Nubia Esmeralda Cantú Sánchez, César Aldahir Flores Gámez, José Antonio Cumpean Morales, Francisco Gael Sustaita Reina, Mauricio Hernández Cepeda, Marco Aurelio Nuño Maganda

Universidad Politécnica de Victoria. Ciudad Victoria, Tamaulipas, México.

2030210@upv.edu.mx, 2030070@upv.edu.mx, 2030367@upv.edu.mx, 2030048@upv.edu.mx, 2030285@upv.edu.mx, mnunom@upv.edu.mx.

Abstract

The recent pandemic has brought with it the search for tools in the learning of technologies, which is why new programs that help to understand the operation of technologically important topics are essential for these times. This paper describes the development of a neural network simulator of the Perceptron algorithm in Python to classify two classes: green and red. The project's main objective is to train the neural network's performance and achieve optimal accuracy in predicting the class to which a new point belongs. The report details the complete implementation process, from data acquisition and preprocessing to parameter configuration and evaluation of new point classification.

Resumen

La reciente pandemia ha traído consigo la búsqueda de herramientas en el aprendizaje de las tecnologías, es por eso que nuevos programas que ayuden a entender el funcionamiento de temas de importancia tecnológica son esenciales para estos tiempos. En este artículo se describe el desarrollo de un simulador de red neuronal del algoritmo de Perceptrón en Python para la clasificación de dos clases: verde y rojo. El objetivo principal del proyecto es entrenar el funcionamiento de la red neuronal y lograr una precisión óptima en la predicción de la clase a la que pertenece un nuevo punto. El informe detalla el proceso completo de implementación, desde la adquisición y preprocesamiento de los datos hasta la configuración de los parámetros y la evaluación de la clasificación de nuevos puntos.

_____

Keywords and phrases: Python, PyQt5, Neuronas, Perceptrón, Predicción, Red Neuronal, Clasificación.

_____

1 Introducción

Como se menciona en [1], las redes neuronales no son una novedad. En 1958 Frank Rosenblatt, psicólogo estadounidense, conceptualizó y trató de construir una máquina que respondiera como la mente humana , llamada Perceptrón. En términos prácticos, las redes neuronales artificiales aprenden mediante ejemplos, de manera similar a las neuronas biológicas. Las entradas externas se reciben, procesan y activan de la misma forma que el cerebro humano.

El presente proyecto tiene como objetivo crear un simulador de una red neuronal basada en el algoritmo de Perceptrón propuesto por Frank Rosenblatt para clasificar en dos clases: verde o rojo. Los datos de estas clases se representan mediante coordenadas en un plano cartesiano y en forma tabular. A través de un proceso de entrenamiento, el simulador es capaz de predecir a qué clase pertenece un punto dado, es decir, realizar una clasificación.

Los algoritmos de Scikit-learn, que es una librería utilizada para la implementación de la red, se combinan y depuran con otras estructuras de datos y aplicaciones externas como Pandas o PyBrain. Con la ayuda de esta librería, se podrán implementar los datos de entrenamiento necesarios para el correcto funcionamiento de la red neuronal.

Es importante mencionar que esta red neuronal cuenta con dos neuronas para la capa de entrada representando las dos clases, una capa oculta que puede tener hasta ocho neuronas y una neurona para la capa de salida la cual tiene la probabilidad de pertenecer a la clase positiva. Como se explica en [1], las capas ocultas de una red neuronal contienen unidades no observables. El valor de cada unidad oculta es alguna función de los predictores, y la forma exacta de la función depende en parte del tipo de red. El presente informe detalla el desarrollo de dicho proyecto de simulador de una red neuronal. En las siguientes secciones, se describirá el proceso completo de implementación, desde la adquisición y preprocesamiento de los datos hasta la configuración de los parámetros de la red y la evaluación de la clasificación de nuevos puntos.

2 Marco Teórico

A continuación, se definen conceptos importantes para la comprensión de este artículo.

2.1. Redes Neuronales Artificiales

Las redes neuronales artificiales son un área de estudio dentro del campo de la inteligencia artificial que ha ganado considerable atención en los últimos años. Están inspiradas en el funcionamiento del cerebro humano y se emplean para simular el procesamiento de información y la toma de decisiones de manera similar a como lo hace el cerebro.

Dichas redes son modelos matemáticos compuestos por un conjunto de unidades interconectadas llamadas neuronas artificiales o nodos. Las cuales están organizadas en capas y se comunican entre sí a través de conexiones ponderadas. La estructura de las redes neuronales puede variar dependiendo del problema a resolver, pero en general constan de una capa de entrada, una o varias capas ocultas y una capa de salida.

El funcionamiento de una red neuronal se basa en la propagación de la información a través de las conexiones entre neuronas. Cada neurona recibe una entrada, realiza un cálculo utilizando

una función de activación y produce una salida. La información fluye desde la capa de entrada, pasando por las capas ocultas, hasta llegar a la capa de salida, donde se obtiene el resultado final.

Una de las principales características de las redes neuronales es su capacidad para aprender a partir de ejemplos. El aprendizaje se realiza mediante algoritmos de entrenamiento que ajustan los pesos de las conexiones entre las neuronas. Dos tipos comunes de aprendizaje en el área son el aprendizaje supervisado, donde se proporciona un conjunto de datos de entrada y sus correspondientes salidas deseadas, y el aprendizaje no supervisado, donde la red encuentra patrones y estructuras en los datos sin necesidad de etiquetas, se puede consultar más información sobre el tema en [3].

2.2. Aprendizaje automático supervisado

El aprendizaje automático supervisado es una rama del aprendizaje automático que se enfoca en la construcción de modelos y algoritmos capaces de aprender a partir de ejemplos etiquetados. Se trata de un enfoque en el que un algoritmo o modelo se entrena utilizando un conjunto de datos que contiene pares de entrada-salida conocidos. El objetivo es encontrar una función que pueda generalizar y realizar predicciones precisas en nuevos datos no vistos [4].

El aprendizaje supervisado se puede aplicar a una amplia gama de problemas. En la clasificación, el objetivo es asignar una etiqueta o categoría a una entrada según las clases conocidas en el conjunto de datos de entrenamiento [5].

2.3. MLPClassifier

El MLPClassifier es un algoritmo de aprendizaje automático supervisado disponible en la librería Scikit-learn que se utiliza para resolver problemas de clasificación utilizando redes neuronales artificiales con múltiples capas. MLP se refiere a Multi-Layer Perceptron, que es un tipo de arquitectura de red neuronal que consta de una capa de entrada, una o más capas ocultas y una capa de salida. Cada capa está compuesta de neuronas, que realizan operaciones matemáticas para procesar la información y transmitirla a la siguiente capa.

El entrenamiento del MLPClassifier se basa en el algoritmo de retropropagación (backpropagation). Durante el entrenamiento, se proporciona al modelo un conjunto de datos de entrenamiento. El algoritmo ajusta los pesos de las conexiones entre las neuronas para minimizar una función de pérdida. Este proceso se repite de manera iterativa hasta que se alcanza la convergencia o se cumple un criterio de parada.

2.4. Algoritmo Perceptrón

Un Perceptrón es una neurona artificial, y, por tanto, una unidad de red neuronal. Este efectúa cálculos para detectar características o tendencias en los datos de entrada.

Según la Perceptrón Learning Rule (regla de aprendizaje del Perceptrón), el algoritmo enseña automáticamente los coeficientes de peso óptimo. El Perceptrón recibe múltiples señales de entrada. Si la suma de las señales supera un umbral determinado, se produce una señal o, por el contrario, no se emite ningún resultado, consulta más información [7].

3 Arquitectura de la Aplicación

Se llevó a cabo la implementación de un sistema de simulación de red neuronal mediante un algoritmo intuitivo utilizando los recursos que brinda el lenguaje de programación orientado a objetos de Python y la librería de interfaz gráfica PyQt5 aprovechando las diversas propiedades que esta provee proporcionando una retroalimentación al usuario.

3.1. Estructura del programa

El algoritmo cuanta con dos clases las cuales son MainWindow y PainterWidget.

La clase MainWindow es la clase principal que define la ventana de la aplicación. Esta hereda de la clase QMainWindow sus funciones y contiene todos los componentes visuales y lógica relacionada con la interfaz de usuario. Está conformada por los siguientes métodos:

- __init__(self): Es el constructor de la clase que inicializa la ventana y configura los componentes.
- open_file(self): Abre un archivo CSV seleccionado por el usuario y muestra sus datos en la tabla.
- validate_csv_format(self, file path): Verifica el formato del archivo CSV para asegurarse de que cumple con los requisitos de columnas y valores.
- read_csv(self, file path): Lee los datos de un archivo CSV y los agrega a la tabla y al dataframe.
- update_graph(self): Actualiza el grafico con los puntos almacenados en self.points.
- load_table(self): Carga los valores del dataframe en la tabla.
- entrenar(self): Entrena la red neuronal utilizando los datos del dataframe y los parámetros seleccionados por el usuario.
- change_next_point_color(self, color): Cambia el color del siguiente punto a agregar en el gráfico.
- on_cell_changed(self, row, column): Actualiza los datos cuando se modifica una celda en la tabla.
- add_row(self): Agrega una nueva fila a la tabla y al dataframe.
- delete_row(self): Elimina una fila de la tabla y del dataframe.
- export_csv(self): Exporta los datos de la tabla a un archivo CSV.
- on_canvas_click(self, event): Maneja el evento de clic en el lienzo del gráfico y agrega puntos en el gráfico.
- is_point_valid(self, x, y): Verifica si un punto a agregar en el gráfico es válido y no se superpone con otros puntos existentes.
- on_color_change(self, index, selected_color=None): Maneja el evento de cambio de color seleccionado en el combobox.
- predecir(self): Realiza la predicción de la clasificación para el punto de prueba agregado y muestra el resultado.
- reset_content(self): Restablece los valores y la interfaz de la aplicación a su estado inicial.

Con ello obtenemos el diagrama de la clase MainWindow, véase en la figura 1.

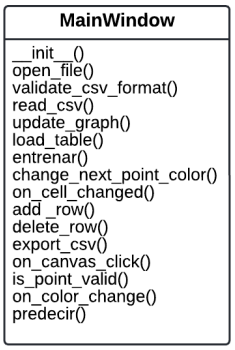


Figura 1. Diagrama de clase MainWindow.

La clase PainterWidget es una clase auxiliar que define un widget personalizado para dibujar la estructura de la red neuronal en forma gráfica. Está conformada por los siguientes métodos:

- __Init__(self): Es el constructor de la clase que inicializa los componentes y variables.
- paintEvent(self, event): Dibuja los círculos y las conexiones de la red neuronal en el lienzo.
- add_neuron(self): Agrega una neurona a la capa oculta de la red neuronal.
- remove_neuron(self): Elimina una neurona de la capa oculta de la red neuronal.

Esto puede ser observado en el diagrama de clase PainerWidget mostrado en la figura 2.

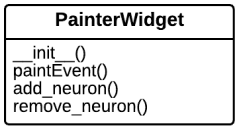


Figura 2. Diagrama de clase PainterWidget.

3.2. Funcionalidad

Una vez iniciado el programa, el usuario tiene la opción de importar un archivo CSV o ingresar los datos manualmente. También puede alternar entre ambas opciones en diferentes pruebas. Los datos ingresados se cargan automáticamente en una tabla, y el usuario puede realizar modificaciones en ellos, como agregar nuevas filas, editar valores existentes o eliminar filas.

Una vez que se han ingresado los datos, se hace clic en el botón "Entrenar" para comenzar el entrenamiento de la red neuronal. Los datos se separan en características y etiquetas. Las características son las columnas del archivo CSV o los datos ingresados manualmente, excepto la última columna, que contiene las etiquetas de clasificación. Luego, los datos se dividen en conjuntos de entrenamiento y prueba utilizando la función train_test_split de sklearn. En este caso, se utiliza una proporción del 80 % para el entrenamiento y el restante 20 % para las pruebas.

A continuación, se crea una instancia de la red neuronal utilizando la clase MLPClassifier de sklearn, que implementa una red neuronal multicapa para clasificación. Se ajustan los parámetros de la red neuronal, como el número de capas ocultas, que en este caso es una, el número de neuronas en cada capa, las cuales son definidas por el usuario solamente en la capa oculta, la función de activación, entre otros datos.

Después de ajustar los parámetros, se entrena la red neuronal utilizando los datos de entrenamiento mediante el método fit de la clase MLPClassifier. Una vez finalizado el entrenamiento, la red neuronal esta lista para hacer predicciones.

El usuario puede agregar un punto de prueba en el gráfico haciendo clic en el lienzo y luego haciendo clic en el botón "Predecir". La posición del punto de prueba se utiliza como entrada para la red neuronal, y se realiza una predicción de clasificación utilizando el método predict de la clase MLPClassifier. El resultado de la predicción se muestra en la etiqueta de la interfaz gráfica.

Es importante destacar que el usuario puede realizar múltiples predicciones según sea necesario. Además, gracias a la función de exportar, el usuario puede guardar los datos que ingresó haciendo clic en el botón "Exportar a CSV." Los datos se guardan en un nuevo archivo CSV en la ubicación seleccionada por el usuario.

3 Resultados

Para comenzar con la parte de los resultados, debemos de mostrar la interfaz inicial es la que se muestra en la figura 3a, la cual cuenta con distintas funciones que se explicaran a lo largo de esta sección. Para iniciar, debemos comenzar a ingresar filas en nuestro dataset, para que esta se dibuje de manera bidimensional en la gráfica. Para hacer esto primero se debe de seleccionar la opción de Agregar fila, y se añadirá una a la fila en donde tendremos que colocar las coordenadas de nuestros dataset, en un intervalo entre 0 y 1, y el color que deseamos que se muestre en el gráfico, ya sea rojo o verde, figura 3b.

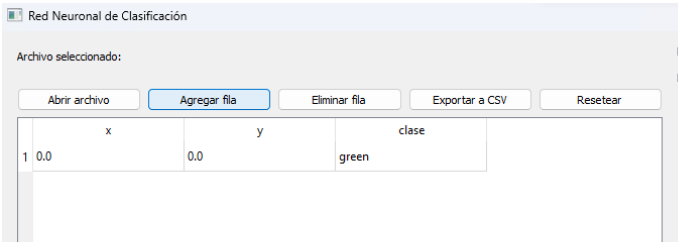
En caso de que en el apartado de color, asignemos un color que no sea rojo o verde, obtendremos un error indicando que debemos de asignar un color valido, ya sea rojo o verde.

Asimismo, en caso de que asignemos un valor al dataset que sea mayor que 1, o menor que 0, es decir, un valor fuera del rango de 0 y 1, como en este caso se ingresó el número 2, obtendremos un error indicando que debemos de asignar un valor valido dentro del rango ya mencionado.

Además de agregar datos al dataset de manera manual a través de la tabla, también podremos asignarlos de manera directa en el gráfico, donde seleccionaremos primeramente el color del punto que deseamos colocar, y una vez seleccionado bastaría dar clic en la posición que deseamos que se muestre en el gráfico, como se muestra en la figura 4a, donde se colocó un punto de color verde, y a la vez este se asignó en la tabla del dataset.

Otra manera es importando un archivo CSV, y para esto daremos clic en el botón de abrir archivo, y en ese momento se nos abrirá una ventana emergente para seleccionar un archivo .CSV que desee importar, como se muestra en la figura 4b, siempre y cuando este tenga datos válidos para nuestro dataset. En caso de que importemos un archivo con datos no compatibles, se nos mostrara un error.

(a) Pantalla inicial.



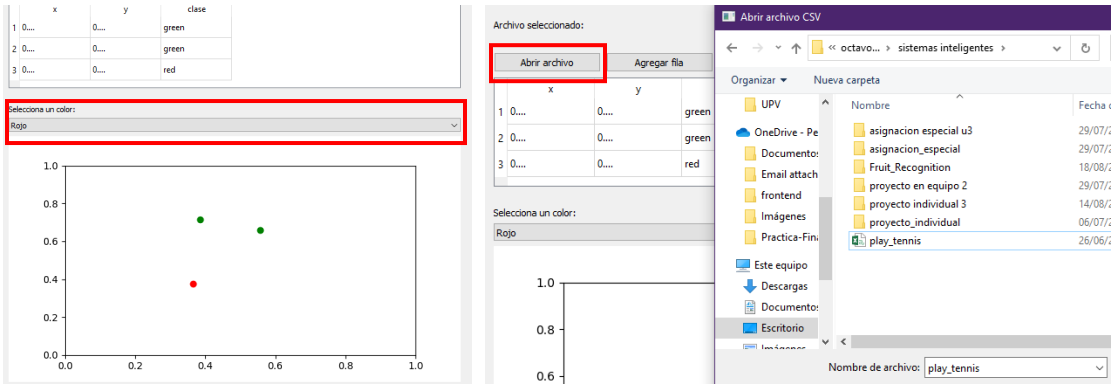
(b) Agregar fila.

Figura 3. Interfaz de la aplicación propuesta, para crear y/o modificar el dataset.

Además de poder asignar filas, también podremos eliminar, dando clic en el botón de Eliminar fila. En la parte de Regularization, podemos elegir uno de los valores que se muestran, siendo el más pequeño de 0.001, y el más grande de 10, siempre y cuando este tenga datos válidos para nuestro dataset. Tenemos también la opción de Activation, en donde se podrá elegir entre la opción de relu, tanh, logistic e identity, como se muestra en la figura 5a. La siguiente opción que podemos configurar sería la de *Learning Rate*, donde podremos elegir alguna de las opciones de constant, invscaling, y adaptive, como se muestra en la figura 5b.

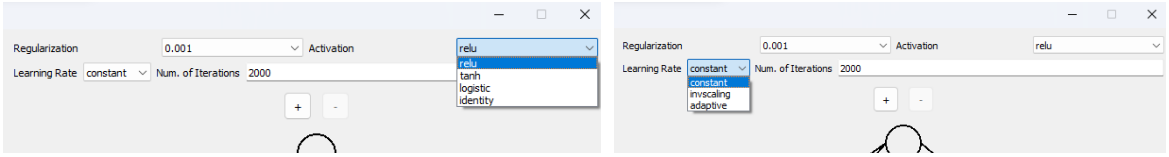
Otra de las opciones que se pueden modificar, sería el valor del número de iteraciones, donde se puede colocar el valor mínimo de 1, y un máximo de 9999, apreciable en la figura 6a, donde en este caso tiene un valor de 2000. Por último, está la configuración de las neuronas, en donde podemos asignar un mínimo de 1, y un máximo de 8. Para poder hacer esto debemos de cliquear la opción de + o -, dependiendo de lo que queremos realizar, figura 6b.

Una vez explicada todas las opciones del programa, pasemos a la parte del entrenamiento, solo bastaría cliquear la opción de “Entrenar”. Cuando cliquemos en Entrenar, se mostrara un aviso que debemos de colocar un punto a clasificar, se debe de cliquear en “ok” para continuar.



(a) Asignar al dataset de manera gráfica (b) Importar archivo .csv.

Figura 4. Interfaz de la aplicación propuesta, donde es posible agregar elementos al dataset e importarlos de un archivo.



(a) Activation. (b) Learning Rate.

Figura 5. Elementos de la interfaz que permiten la selección de la función de activación y establecer el *Learning rate.*

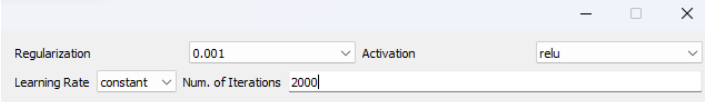
Continuando con el procedimiento, se debe de cliquear en el grafico en la posición que deseamos colocar el punto a clasificar. Este punto se mostrara de color azul, y se bloquearan la mayoría de opciones, como se aprecia en la figura 7, donde una vez colocado el punto deseado, bastaría cliquear en la opción de predecir. Una vez realizado dicha acción, se mostraran los resultados en la parte derecha de la interfaz, mostrándolos de manera gráfica, y de manera escrita, además de mostrar la clase predicha, ya sea red o green.

Por último, tenemos 2 opciones más, resetear el programa, o exportarlo en un archivo CSV. En caso de que se resetee el programa, se mostrara la interfaz inicial como en la figura 3a, y en caso de querer exportarlo, se debe de cliquear la opción correspondiente, y se abrirá el explorador de archivos para buscar donde se desea guardarlo, se le asigna un nombre al archivo, y por último bastaría cliquear en “Guardar”.

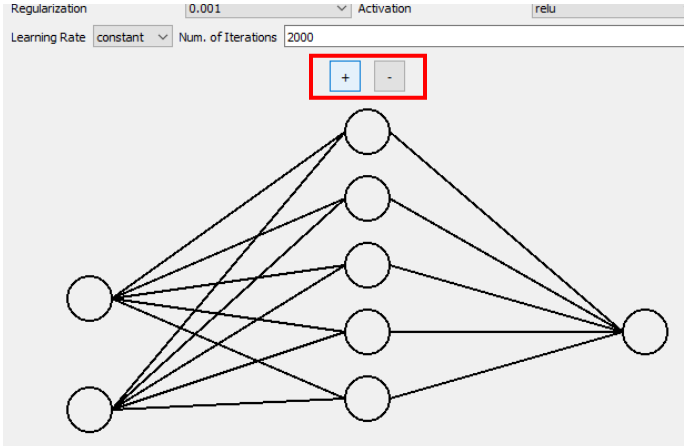
5 Conclusiones y trabajo a futuro

Actualmente las aplicaciones en línea son lo más común, sin embargo no todas las personas tiene el acceso fácil a internet, por lo que la propuesta del simulador descrito en este artículo ha sido propuesto como una aplicación de escritorio el cual puede funcionar de forma independiente. Otras

aplicaciones en línea demostraron que si bien ya existen simuladores parecidos al propuesto no tienen la facilidad de personalización que el creado para fines de demostración en este artículo. Se pretende además con este programa ayudar a entender el funcionamiento de las redes neuronales por lo que la unión de distintas funcionalidades concentradas en una sola interfaz hace el uso de esta aplicación más efectivo que el de otras.



(a) Número de iteraciones.



(b) Neuronas.

Figura 6. Elementos de la interfaz que permiten realizar el entrenamiento en base a los parámetros establecidos.

En este proyecto, se propuso una solución para crear una aplicación de escritorio de un simulador de redes neuronales que pudiera reconocer y clasificar diferentes clases utilizando pesos aleatorios. Se desarrolló un programa utilizando Python, programación orientado a objetos, y se usaron varias bibliotecas, como PyQt5 para la interfaz gráfica y las animaciones, y Scikit-learn para la demostración de la red neuronal y el cálculo de los pesos.

A lo largo del proyecto, se abordó la problemática de clasificar puntos utilizando redes neuronales y se exploraron diferentes enfoques para mejorar el rendimiento. Se decidió utilizar la biblioteca Scikit-learn y el algoritmo ADAM para obtener mejores resultados.

Se identificaron áreas de mejora para el sistema. En cuanto al aspecto visual, se reconoce la necesidad de mejorar la interfaz para hacerla más agradable y atractiva para el usuario. Además, en términos de funcionalidad, se encontró que el sistema tiene un número fijo de iteraciones, lo que podría limitar su capacidad de procesar grandes cantidades de información de manera confiable.

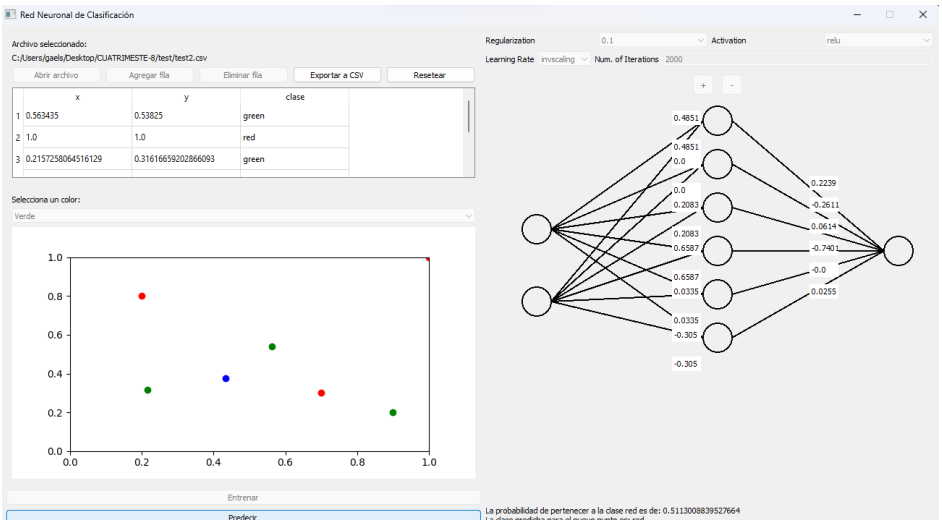


Figura 7. Elementos de la interfaz que permiten la clasificación de nuevos elementos.

En conclusión, la combinación de simuladores de redes neuronales y el uso de PyQt5 demostró ser una plataforma poderosa para la creación de un simulador visualmente atractivo.

En cuanto al trabajo futuro, se sugiere continuar mejorando la interfaz gráfica para ofrecer una experiencia más agradable al usuario. Además, se puede explorar la posibilidad de implementar técnicas de entrenamiento más avanzadas y optimizaciones en la red neuronal para mejorar su precisión y capacidad de procesamiento de datos.

Referencias

[1] TIBCO, "¿qué es una red neuronal?," 2020. https://www.tibco.com/es/reference-center/what-is-a-neural-network

[2] U. de Alcalá, "¿qué es scikt-learn?," 2020. https://www.master-data-scientist.com/ scikit-learn-data-science/

[3] IBM, "Redes neuronales". 2020. https://www.ibm.com/mx-es/topics/neural-networks.

[4] A. Z., A. Casari. Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists. O'Reilly Media. 2018.

[5] S. R., V. Mirjalili, Python Machine Learning. Packt Publishing, 2015.

[6] W. McKinney. Python for Data Analysis. O'Reilly Media. 2012.

[7] Team, D. (2022). "Perceptrón : ¿qué es y para qué sirve?". Formation Data Science | DataScientest.com. https://datascientest.com/es/Perceptrón-que-es-y-para-que-sirve#:~:text=El%20perceptr%C3%B3n%20efect%C3%BAa%20c%C3%A1lculos%20para,de%20una%20serie%20de%20datos.

Implementación de una interfaz gráfica basada en PyQt5 para el algoritmo de K-vecinos más cercanos.

Implementation of a PyQt5-based graphical interface for the K-nearest neighbors algorithm.

Yanel Azucena Mireles Sena, Sonia Lizbeth Muñoz Barrientos, Kency Marisol Saldaña Martinez, Vanessa Itzaiana García Cervantes, Jorge Luis Charles Torres,

Universidad Politécnica Victoria, Ingeniería en Tecnologías de la Información

2030055@upv.edu.mx, 2030114@upv.edu.mx, 203002@upv.edu.mx, 2030208@upv.edu.mx, 2030340@upv.edu.mx.

Abstract

The K-Nearest Neighbors Demo project, which focuses on the implementation and demonstration of the K-Nearest Neighbors algorithm in the context of machine learning. The goal of the project is to provide a practical understanding of the algorithm and its application in classification and regression problems. An intuitive graphical interface was developed to allow users to explore the operation of the algorithm and relevant code examples will be provided.

Resumen

El proyecto K-Nearest Neighbors Demo, el cual se enfoca en la implementación y demostración del algoritmo de vecinos más cercanos (K-Nearest Neighbors) en el contexto del aprendizaje automático. El objetivo del proyecto es brindar una comprensión práctica del algoritmo y su aplicación en problemas de clasificación y regresión. Se desarrolló una interfaz gráfica intuitiva para permitir a los usuarios explorar el funcionamiento del algoritmo y se proporcionarán ejemplos de código relevante.

Keywords and phrases: K-nearest Neighbors, K-NN, Aprendizaje Automático, Dataset, Neighbors, K number.

1 Introducción

El presente artículo tiene como propósito presentar el proyecto denominado K-Nearest Neighbors Demo, el cual se enfoca en la implementación y demostración del algoritmo de vecinos más cercanos (K-Nearest Neighbors, K-NN en inglés) mediante el uso de un conjunto de herramientas y librerías de Python. En un mundo moldeado por la reciente pandemia, las dinámicas educativas y de adquisición de conocimientos han experimentado cambios significativos. La necesidad de adaptarse a nuevas modalidades de aprendizaje ha impulsado la búsqueda de herramientas complementarias que enriquezcan la comprensión de tecnologías clave.

La situación global no solo ha desafiado los métodos educativos tradicionales, sino que también ha instigado una reevaluación de cómo los individuos pueden asimilar eficazmente conceptos

complejos, como los algoritmos de aprendizaje automático. En este contexto, el algoritmo K-NN se destaca como una herramienta de gran utilidad en el aprendizaje supervisado, capaz de realizar clasificaciones y predicciones precisas a través de la evaluación de la proximidad entre datos.

Surge así el proyecto K-Nearest Neighbors Demo como respuesta a la necesidad de una herramienta práctica que permita explorar y comprender de manera efectiva el algoritmo de vecinos más cercanos. A lo largo de este artículo, no solo se profundizará en el funcionamiento del algoritmo y su implementación, sino que también se abordará la relevancia de esta iniciativa en el actual panorama educativo y tecnológico. Ejemplos de código pertinentes serán compartidos y los resultados de su aplicación en diversos conjuntos de datos serán analizados minuciosamente.

Mediante este artículo, se busca fomentar una comprensión sólida y rigurosa del algoritmo K-NN y su capacidad para potenciar el aprendizaje de tecnologías, especialmente en un contexto en el que la adaptabilidad y las herramientas complementarias son más valiosas que nunca.

2 Marco Teórico y Estado del Arte

El algoritmo K-Nearest Neighbors (K-NN) es un método de aprendizaje supervisado utilizado en problemas de clasificación y regresión. En este algoritmo, se busca predecir la clase o valor de una nueva instancia intensa en las características de las instancias vecinas más cercanas en un conjunto de datos de entrenamiento.

2.1. Algoritmo K-nearest neighbors

El algoritmo K-NN se basa en la idea de que las instancias similares pueden tener a pertenecer a la misma clase o tener valores similares. Por lo tanto, se utiliza una medida de distancia para encontrar las instancias más cercanas en el espacio de características. La medida de distancia más utilizada es la distancia euclidiana, aunque también se pueden utilizar otras medidas de distancia. El proceso de trabajo del algoritmo K-NN se puede describir en los siguientes pasos:

1. Preprocesamiento de datos: En primer lugar, es importante realizar un preprocesamiento de los datos, lo que implica la limpieza de datos, la normalización y la eliminación de características irrelevantes o ruido, si es necesario. Esto ayuda a mejorar la precisión del algoritmo y a eliminar posibles sesgos.
2. Definir el valor de "K": El siguiente paso es seleccionar el valor de "K", que representa el número de vecinos más cercanos que se utilizarán para hacer una predicción. El valor de "K" se selecciona según las características del conjunto de datos y el problema en cuestión.
3. Calcular distancias: Para encontrar los vecinos más cercanos, se calcula la distancia entre el nuevo punto de datos y todos los puntos de entrenamiento existentes. Las distancias más comúnmente utilizadas son la distancia euclidiana o la distancia Manhattan, aunque se pueden utilizar otras métricas según el problema.
    a. La distancia euclidiana es la longitud de la línea recta que conecta dos puntos en un espacio euclidiano. Imagina dos puntos en un plano cartesiano: la distancia euclidiana entre ellos es la longitud de la línea recta que los conecta.
    b. La distancia Manhattan o longitud Manhattan (geometría del taxi) nos dice que la distancia entre dos puntos es la suma de las diferencias absolutas de sus

coordenadas. Es decir, es la suma de las longitudes de los dos catetos del triángulo rectángulo.

4. Selección de vecinos: Después de calcular las distancias, se seleccionan los "k" vecinos más cercanos según la distancia calculada. Estos vecinos más cercanos pueden considerarse como votos en el caso de clasificación o como contribuciones ponderadas en el caso de regresión.
5. Realizar la predicción: Una vez que se han seleccionado los vecinos más cercanos, se utiliza su etiqueta (en el caso de clasificación) o su valor (en el caso de regresión) para realizar una predicción para el nuevo punto de datos. En el caso de clasificación, se puede utilizar el voto mayoritario para determinar la clase del nuevo punto. En el caso de regresión, se puede tomar el promedio de los valores de los vecinos más cercanos.
6. Evaluar el rendimiento: Finalmente, se evalúa el rendimiento del algoritmo K-NN utilizando técnicas de validación cruzada u otras medidas de evaluación adecuadas, como la precisión, el error cuadrático medio, entre otros.

El valor de K en K-NN se refiere al número de instancias vecinas más cercanas que se utilizan para predecir la clase o valor de la nueva instancia. El valor de K se selecciona de antemano y puede afectar significativamente la precisión del modelo. Un valor de K pequeño puede hacer que el modelo sea demasiado sensible a ruido o variaciones en los datos, mientras que un valor de K grande puede hacer que el modelo sea demasiado generalizado y pierda detalles importantes.

2.2. Estado del Arte

Se han desarrollado diversas variantes y mejoras del algoritmo K-NN a lo largo del tiempo. Esto ha permitido el desarrollo de distintas herramientas parecidas a la presentada en este artículo, algunas de las cuáles fueron revisadas y analizadas para encontrar el valor agregado de nuestra herramienta, dichos proyectos se detallan a continuación:

1. k-Nearest Neighbor (kNN) Classifier - WOLFRAM Demonstrations Project [1]: Este es un proyecto utilizado para abordar problemas de clasificación en el ámbito del aprendizaje automático. Su función principal es predecir la clase de salida (representada como 0 o 1) en función de las entradas proporcionadas (características o variables de entrada).

   Sin embargo presenta algunas limitaciones a comparación a nuestra herramienta:

   - El valor de K sólo puede ser de 1 a 17.
   - Sólo puede usar 2 clases, que se representan en rojo y verde.
   - Los puntos no pueden ser asignados a gusto del usuario ni ser modificados.

2. KNN Demo - CodePen Home [2]: Esta es una herramienta comercial especializada y enfocado a entornos de programación. Su visualización gráfica muestra a través de líneas las uniones con los vecinos más cercanos. Para aprovechar al máximo la herramienta se requieren conocimientos en HTML, Javascript y CSS para el control de cada vecino y clase.

   Las desventajas frente a nuestra herramienta son las siguientes:

   - Tiene una versión gratuita con limitaciones muy estrictas como la cantidad de puntos a agregar y las clases.

- Versión de paga para acceso ilimitado.
- Su nivel de uso es complejo.
- No es didáctico.

Nuestra herramienta se destaca por las siguientes ventajas:

- **No utiliza internet:** A diferencia de los otros demos, nuestra herramienta funciona sin conexión a Internet, lo que la hace más versátil y adecuada para entornos con conectividad limitada.
- **Carga de archivos CSV:** Permite a los usuarios cargar archivos CSV, facilitando la importación de datos desde fuentes externas de manera rápida y sencilla.
- **Flexibilidad en la carga de puntos:** Los usuarios pueden agregar tantos puntos como deseen en nuestra herramienta, lo que brinda flexibilidad para trabajar con conjuntos de datos de diferentes tamaños y complejidades.
- **Interfaz didáctica:** Nuestra herramienta está diseñada para ser didáctica, lo que la hace accesible incluso para aquellos sin conocimientos profundos de programación.

Es importante señalar que, aunque nuestra herramienta presenta algunas limitaciones, estas son mínimas en comparación con las ventajas que ofrece:

- **Limitaciones en la complejidad del modelo:** Dada su orientación didáctica, nuestra herramienta puede no ser adecuada para tareas extremadamente complejas que requieren modelos de aprendizaje automático altamente especializados. Sin embargo, para la mayoría de las aplicaciones, su capacidad es suficiente.

En resumen, nuestra herramienta supera a las mencionadas anteriormente debido a su facilidad de uso, versatilidad y capacidad para adaptarse a diversas necesidades de los usuarios, y las desventajas son mínimas en relación con sus ventajas.

3 Arquitectura de la Aplicación

3.1 Visión General

La arquitectura de la aplicación está diseñada para ser modular y fácilmente extensible. La aplicación se desarrolló utilizando el lenguaje de programación Python y se apoya en varias bibliotecas externas para llevar a cabo tareas específicas.

3.2 Componentes Principales

- Interfaz de Usuario (UI): Utilizamos PyQt5 para construir la interfaz gráfica de usuario, que incluye botones para seleccionar la clase, un cuadro combinado para seleccionar el color de los puntos, y un canvas para dibujar los puntos y visualizar los resultados del algoritmo K-NN.
- Algoritmo K-NN: Implementado usando la biblioteca scikit-learn (sklearn). Este componente se encarga de la clasificación de los puntos en el canvas.
- Gestión de Datos: Utilizamos pandas para la importación y exportación de datos a y desde archivos CSV.
- Visualización: Matplotlib se utiliza para dibujar los puntos y las regiones en el canvas.

3.3 Flujo de la Aplicación

- Inicialización: Al iniciar la aplicación, se crea una ventana que contiene todos los elementos de la UI.
- Selección de Clase y Color: El usuario selecciona una clase y un color utilizando los botones y el cuadro combinado.
- Dibujo de Puntos: El usuario puede hacer clic en el canvas para añadir puntos, que se clasificarán en tiempo real.
- Aplicación de K-NN: Al añadir un nuevo punto o cambiar el valor de k, se aplica el algoritmo K-NN para clasificar los puntos.
- Importación/Exportación de Datos: El usuario tiene la opción de importar o exportar los puntos y sus clases a un archivo CSV.

3.4 Bibliotecas Externas

- PyQt5: Para la creación de la interfaz gráfica de usuario.
- scikit-learn (sklearn): Para implementar el algoritmo K-NN.
- pandas: Para la gestión de datos.
- NumPy: Para operaciones matemáticas.
- Matplotlib: Para la visualización de datos.
- SciPy: Para cálculos adicionales, como la distancia entre puntos.

4 Funcionalidad

El conjunto de datos se puede definir de dos formas diferentes. Una de ellas es a través de un archivo CSV generado por los integrantes del equipo, y la otra es colocando los puntos de manera aleatoria. El programa se inicia mostrando una interfaz gráfica que incluye una gráfica vacía y varios componentes interactivos. El usuario tiene dos opciones para ingresar los datos: importar un archivo CSV o agregarlos manualmente.

Si el usuario elige importar un archivo CSV, puede hacerlo haciendo clic en el botón Importar CSV. A través de un cuadro de diálogo, el usuario selecciona el archivo CSV deseado y el programa carga automáticamente los datos numéricos en la gráfica. Los puntos se asignan a las clases correspondientes según los colores especificados en el archivo. En caso de que el usuario prefiera ingresar los datos manualmente, simplemente debe hacer clic en la gráfica en la ubicación deseada. Al hacerlo, se agrega un punto con la clase y el color seleccionados. El usuario puede seleccionar la clase haciendo clic en el botón etiquetado como Clase X, donde X representa el número de clase. Todos los puntos agregados posteriormente se asignan a esa clase y se les dará el color asociado.

Cada vez que se agrega un punto a la gráfica, el programa realiza automáticamente el algoritmo K-NN para clasificar los puntos existentes y los nuevos puntos agregados.El sistema utiliza el número de vecinos seleccionado (solo números impares) para realizar la clasificación.

Los puntos se muestran en la gráfica con los colores asignados según la clasificación realizada. Además, el usuario puede realizar predicciones haciendo clic en cualquier lugar de la gráfica. Utilizando el algoritmo K-NN entrenado, el programa realizará una predicción de clasificación del punto seleccionado y mostrará el resultado en la interfaz gráfica. El usuario también tiene la opción

de exportar los datos de la gráfica a un archivo CSV haciendo clic en el botón Exportar CSV. Esto permite guardar los puntos y sus clasificaciones en un archivo para futuros usos.

5 Resultados

En esta sección se proporcionará una explicación detallada sobre cómo utilizar el programa. Al ejecutar el programa, se presentará la interfaz inicial, que se muestra en la figura 1. Esta interfaz ofrece varias funcionalidades que se describirán a continuación de manera exhaustiva**.**

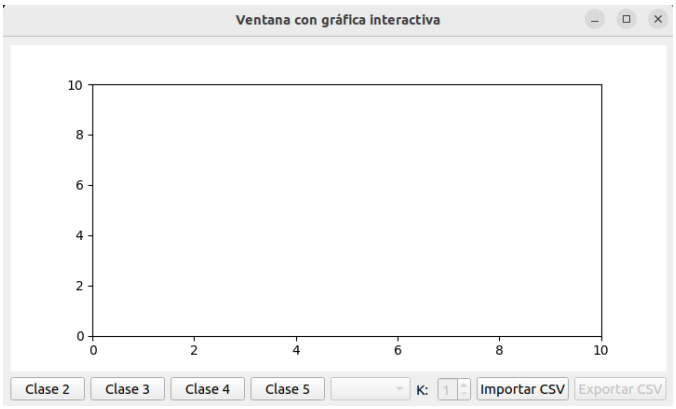
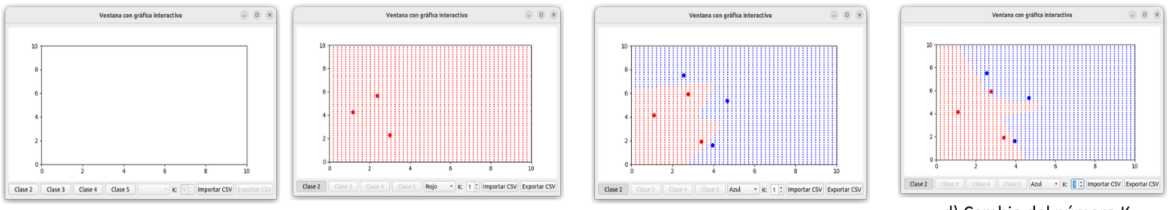


Figura 1. Pantalla de inicio que se muestra al compilar el programa.

Para comenzar, se ofrecen dos opciones al usuario. La primera opción es importar un archivo CSV, la cual se activa al hacer clic en el botón Importar CSV. Al seleccionar esta opción, el sistema abrirá un explorador de archivos que permitirá al usuario buscar y seleccionar el archivo CSV deseado. Una vez seleccionado, el programa procesa el archivo y mostrará en la gráfica los puntos y colores correspondientes a los datos contenidos en el archivo. La segunda opción es ingresar los puntos manualmente en la gráfica. Para ello, se debe seleccionar una clase y su respectivo color. Después de seleccionar la clase, se desbloqueará un menú desplegable con los colores disponibles que corresponden al mismo número de clases seleccionado. El usuario debe elegir el color deseado en el menú desplegable. A continuación, simplemente se debe hacer clic dentro de la gráfica en las posiciones deseadas para agregar los puntos. El sistema aplicará automáticamente el algoritmo de clasificación correspondiente para asignar una clase a cada punto en la gráfica.

Se realizaron diversas tipos de pruebas para verificar el funcionamiento del programa, a continuación se muestran unas de ellas:

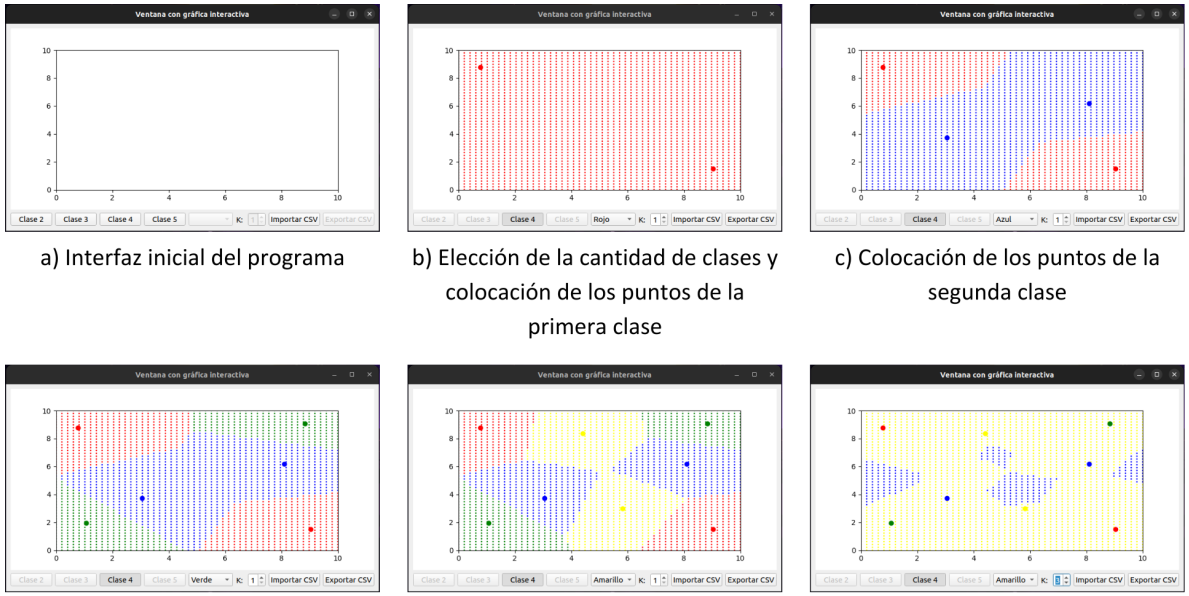
- Prueba 1: esta prueba fue realizada con dos clases por lo cual solo se tendrá el color rojo y azul, los puntos fueron agregados manualmente dentro de la gráfica de una forma aleatoria, que se muestra la secuencia del proceso de la prueba en la figura 2.



a) Interfaz inicial del programa b) Selección del número de clases y colocación de la primera clase c) Colocación de la segunda clase d) Cambio del número K

Figura 2. Visualización del proceso de la prueba número 1.

- Prueba 2: esta prueba fue realizada con cuatro clases por lo cual solo se tendrá el color rojo, azul verde y amarillo, los puntos fueron agregados manualmente por el usuario dentro de la gráfica de una forma aleatoria, se muestra la secuencia del proceso de la prueba en la figura 3.

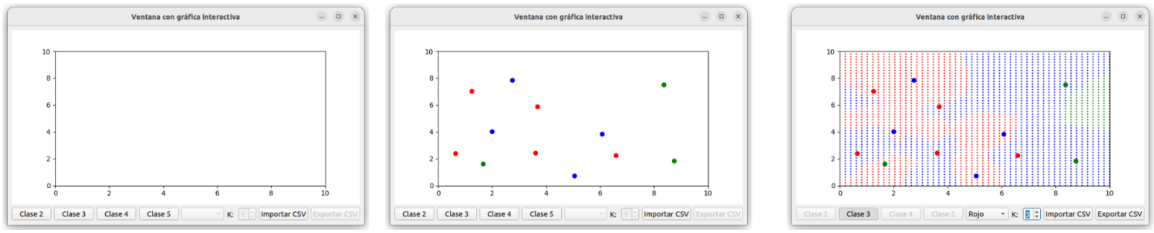


a) Interfaz inicial del programa b) Elección de la cantidad de clases y colocación de los puntos de la primera clase c) Colocación de los puntos de la segunda clase

d) Colocación de los puntos de la tercera clase e) Colocación de los puntos de la cuarta clase f) Cambio del número K

Figura 3. Visualización del proceso de la prueba número 2.

A continuación, se muestra un ejemplo importado utilizando un archivo CSV en la figura 4. Es importante destacar que al importar el CSV, únicamente se marcan los puntos y colores, y es necesario indicar la clase y el valor de K para que se pinten los puntos de fondo. K representa el número de vecinos considerados en el algoritmo de clasificación K-NN es necesario que el usuario elija previamente una clase en la interfaz gráfica. Una vez seleccionada la clase, se habilitará automáticamente un cuadro de selección para K. Dentro de este cuadro de selección, el usuario encontrará una lista de valores predefinidos de K, donde deberá elegir uno de los números impares disponibles. Esta restricción de números impares se implementa para evitar que el algoritmo entre en una una indeterminación de hacia qué color ir.



a) Interfaz inicial del programa b) Importación del archivo CSV c) Selección del número de clase y el número K

Figura 4. Visualización del proceso de un archivo importado CSV.

En caso de que el usuario haga clic en la gráfica sin haber seleccionado previamente una clase, el sistema mostrará una advertencia en forma de mensaje emergente indicando que debe seleccionar una clase antes de agregar puntos. Esta advertencia tiene como objetivo recordar al usuario la importancia de especificar la clase a la que pertenecerán los puntos que está agregando. En la figura 5 se muestra un ejemplo de cómo se ve la advertencia en la interfaz gráfica.

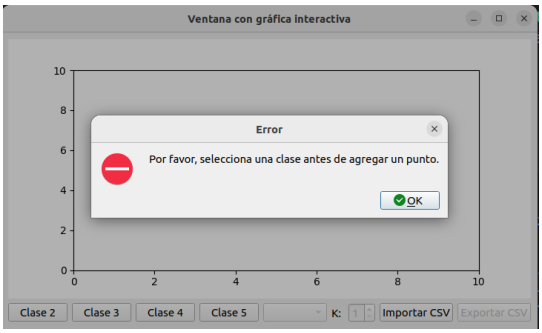


Figura 5. Mensaje de error.

Para reiniciar el programa y poder ingresar una nueva clase, simplemente se debe presionar nuevamente el botón de la clase seleccionada. Al hacerlo, el programa se reiniciará, borrando los puntos previamente colocados en la gráfica y desbloqueando los botones de las clases para permitir al usuario ingresar o importar nuevos datos. Esta funcionalidad brinda flexibilidad al usuario, ya que puede realizar múltiples iteraciones y pruebas sin tener que cerrar y volver a abrir el programa.

6 Conclusiones y trabajo a futuro

El desarrollo de esta aplicación demuestra el potencial significativo de la tecnología en el ámbito educativo, especialmente en la enseñanza de algoritmos de aprendizaje automático como K-NN. La interfaz, diseñada para ser intuitiva y fácil de usar, permite a los usuarios interactuar con el algoritmo de una manera más comprensible. La integración de diversas bibliotecas de Python, como scikit-learn, pandas y Matplotlib, resalta la eficiencia y flexibilidad que las herramientas de código abierto pueden ofrecer en el desarrollo de aplicaciones educativas.

El sistema presentado en este documento podría tener diferentes utilidades en diversas áreas, por ejemplo en el campo de la minería de datos se pudiera emplear para clasificar y agrupar datos, esto se realizaría a través de la identificación de patrones y tendencias en grandes conjuntos de datos,

esto a su vez podría ser útil en áreas como el análisis de mercado, la detección de fraudes, la segmentación de clientes, entre otros.

Otro campo donde probablemente tenga cabida es en la medicina, se pudiera utilizar para diagnosticar enfermedades mediante el análisis de datos médicos, una muestra de esto sería para identificar a pacientes con riesgo de desarrollar una determinada enfermedad en función de su historial médico y otros factores.

6.1 Trabajo a futuro

El presente trabajo ha demostrado la eficacia de la combinación del algoritmo K-Nearest Neighbors (KNN) con una interfaz gráfica basada en PyQt5, permitiendo a los usuarios visualizar y experimentar con la clasificación de datos de una manera interactiva y comprensible. Sin embargo, hay diversas áreas y aspectos que pueden ser mejorados y extendidos en futuros trabajos.

Por ejemplo, aunque KNN es un algoritmo simple y eficaz para ciertas aplicaciones, puede ser ineficiente para conjuntos de datos muy grandes. Sería relevante investigar y, eventualmente, implementar estructuras de datos como árboles KD o algoritmos aproximados de KNN para mejorar el rendimiento en tiempo real. Además, aprovechando la potencia de PyQt5, sería posible implementar una funcionalidad donde los usuarios puedan agregar, eliminar o modificar puntos en el espacio de características y observar cómo cambian las clasificaciones en tiempo real.

También la plataforma podría adaptarse para incluir otros algoritmos de machine learning, brindando a los usuarios una herramienta más completa y educativa sobre diferentes técnicas de clasificación. Incluso, enfocándose más a fondo en la experiencia del usuario, se podría mejorar la adaptabilidad y personalización de la interfaz para que pueda ajustarse a diferentes aplicaciones y preferencias de usuario. Esto podría incluir esquemas de colores, herramientas de anotación y modos de visualización.

El potencial de combinar herramientas educativas y técnicas de machine learning es vasto, y esperamos que este proyecto sirva como un punto de partida inspirador para futuras investigaciones y desarrollos en el campo.

En el contexto del rápido crecimiento de las aplicaciones móviles en la educación, una dirección futura para este proyecto podría ser su adaptación para dispositivos móviles. Además, se podrían incorporar más algoritmos y funcionalidades para enriquecer la experiencia del usuario. Otras mejoras podrían incluir un sistema de evaluación y retroalimentación para medir la eficacia de la aplicación, así como características que permitan la colaboración en tiempo real entre usuarios, fomentando así el aprendizaje colaborativo. Cabe mencionar que la versión actual del sistema cuenta con aspectos a mejorar, tal vez el más sobresaliente es la animación que se realiza al colorear el fondo de los puntos colocados por el usuario, en una versión anterior el fondo se coloreaba mejor, pero requería muchos recursos de la computadora así que se optó por utilizar una especie de cuadrícula para optimizar el proceso de animación, en una versión futura se podría perfeccionar esta animación con el propósito de observar de mejor manera los resultados.

Referencias

[1] Wolfram Demonstrations Project. (s.f.). K-Nearest Neighbor (KNN) Classifier. Demonstrations - Wolfram. URL: https://www.demonstrations.wolfram.com/KNearestNeighborKNNClassifier/

[2] Gangtao. (s.f.). KNN - Demo. CodePen. URL: https://codepen.io/gangtao/pen/PPoqMW

[3] F. Sabry, “K nearest neighbor algorithm: Fundamentals and applications,” 2023.

[4] C. M. P. Pertuz, “Aprendizaje automático y profundo en python,” 2022.

[5] C. P. Lóopez, “Sistemas de aprendizaje automático,” 2022.

[6] E. Alpaydın, “Introduction to machine learning,” 2010.

[7] S. Dark, “Aprendizaje automático,” 2019.

[8] P. H. T. Cover, “Nearest neighbor pattern classification,” IEEE Transactions on Information Theorys, vol. 13, pp. 21–27, jan 1967.

[9] L. P. R. L. J. Krajewski, “Administración de operaciones,” 2000.

[10] P. G. P. Bruce, A. Bruce, “Estadística práctica para ciencia de datos con r y python,” 2022.

[11] I. M. D. D. A. F. Isabel, “Ciencia de datos para la ciberseguridad,” 2020.

[12] T. Cover and P. Hart, “k-nearest neighbors,” Pattern Analysis and Machine Intelligence, IEEE Transac-tions on, vol. 13, no. 3, pp. 314–328, 1986.

Detección de lenguaje misógino en medios sociales en español utilizando Transformers

Detection of misogynistic language in social media in Spanish using Transformers

Ángel Oswaldo Vázquez Benito, Mario Andrés Paredes Valverde, María del Pilar Salas Zárate
División de Estudios de Posgrado e Investigación, Instituto Tecnológico Superior de Teziutlán, Fracción l y ll SN, 73960, Teziutlán, Pue., México.
m21te0021@teziutlan.tecnm.mx, mario.pv@teziutlan.tecnm.mx, maría.sz@teziutlan.tecnm.mx.

Abstract

The use of social networks has been a means of exposure for freely hateful conduct, including the attack on women through comments with misogynistic content. The use of machine learning and artificial intelligence seek technological alternatives in the creation of new models capable of identifying this type of phenomenon in social networks. In this work, an effort is made to identify misogynistic behaviors from a classification model based on Transformers with an accuracy of 0.76 and an F-measure of 0.84, trained with a dataset built from scratch using data obtained from Facebook and Twitter in Spanish.

Resumen

El uso de las redes sociales ha sido un medio de exposición para conductas de odio de forma libre incluyendo el ataque a mujeres a través de comentarios con contenido misógino. El uso del aprendizaje automático y la inteligencia artificial buscan alternativas tecnológicas en la creación de nuevos modelos capaces de identificar este tipo de fenómenos en redes sociales. En este trabajo se hace un esfuerzo por identificar conductas misóginas a partir de un modelo de clasificación basado en Transformers con una precisión de 0.76 y una medida-F de 0.84, entrenado con un dataset construido desde cero usando datos obtenidos de Facebook y Twitter en español.

Keywords and phrases: Lenguaje Misógino, Transformers, BERT.

1 Introducción

Con el paso del tiempo las redes sociales han permitido establecer comunicación desde cualquier parte del mundo, sin embargo, el objetivo inicial de ellas se ha ido distorsionando al encontrar contenido que no solo fomenta una sana comunicación, sino que también se puede identificar la difusión de ideas, lemas, actitudes e incluso conductas a través de Internet que menosprecian,

atacan y humillan a otras personas. Este tipo de conductas suele ser dirigida a grupos concretos por motivos de raza, género, sexualidad o religión.

Según cifras del INEGI [1] se estima que en México la población usuaria de internet que fue víctima de ciberacoso aumentó de 21.0 % en 2020 a 21.7 % en 2021, para el año 2021 el 22.8% de mujeres que usan internet manifestaron haber sido víctimas de ciberacoso.

La misoginia hasta hace algunos años [2] pasaba desapercibida en chistes, bromas, comentarios, etc., pues resultaban ser "sutiles" para la sociedad y no se dimensionaba el impacto que estos pudieran tener sobre el grupo afectado, pero con el uso de redes sociales se han visto expuestos.

La importancia de proyectos dentro del área tecnológica tiene como objetivo el desarrollo de modelos predictivos que servirán como base para el desarrollo de herramientas computacionales o bien para investigaciones posteriores dentro de este ámbito que aporten al avance de la detección de conductas de odio tal como la misoginia en diferentes niveles.

En este trabajo se encuentra una descripción de conceptos que ayudan a la comprensión del tema y una revisión del estado del arte en trabajos recientes que se relacionan con alguna de las vertientes de este proyecto así mismo se da a conocer los pasos para la obtención de un conjunto de datos y sus características que forman el pilar para la construcción de un modelo capaz de detectar contenido misógino así como su evaluación y pruebas que se describen a lo largo de este artículo, finalizando así con las conclusiones y algunas observaciones que podrían mejorar los resultados a futuro.

2 Marco teórico y estado del arte

2.1 Marco teórico

En esta sección se proporciona la definición de algunos conceptos que resultan como factor clave para la correcta comprensión de este artículo.

2.1.1 Lenguaje misógino

Como muchos autores coinciden el objetivo de la misoginia [2] expresada a través del lenguaje es la deshumanización de la mujer y esto se basa en la creencia de que la mujer es inferior al hombre, lo cual ha sido justificado desde diferentes perspectivas; religiosas, biológicas, seudocientíficas e incluso políticas. Actualmente desde la perspectiva de género la misoginia se produce cuando un hombre atenta contra la dignidad, corporeidad y salud mental de una mujer, o al menos siempre se ha tenido la tendencia a transpolar este término, sin embargo, la misoginia no es únicamente lo anterior descrito, es también cualquier conducta de odio contra las mujeres, en donde no necesariamente el autor de dichas conductas debe ser un varón como en el mayor de los casos se tiende a generalizar.

Como en otros discursos de odio, la misoginia se origina en medios digitales cuando, día a día, se favorece su proliferación y la permanencia de una cultura de desigualdad y violencia principalmente sustentada por razones de género, en la que las mujeres son discriminadas y violentadas de manera impune. [3] El lenguaje misógino sucede cuando se codifica la agresión contra las mujeres a través de palabras u otras construcciones lingüísticas. Expresiones en Twitter como "una perra menos" son

misóginas debido a que se refieren a una mujer de manera despectiva y denigrante, es por ello que publicar este tipo de expresiones en las redes sociales es violencia activa.

2.1.2 Aprendizaje automático

El aprendizaje automático [4] puede ser entendido como la rama de la inteligencia artificial (IA) que es empleado en el desarrollo de sistemas que aprenden o mejoran el rendimiento, en función de los datos de los cuales se alimenta. La Inteligencia artificial el termino para referirse a sistemas o máquinas que imitan la inteligencia humana. El aprendizaje automático y la IA suelen ser términos directamente en relación, aunque se resalta la importancia de jerarquía puesto que, aunque todo aprendizaje automático es IA, no toda la IA es aprendizaje automático.

En la actualidad existe una gran variedad de aplicaciones que emplean técnicas de aprendizaje automático en los campos de las ciencias sociales y de la comunicación. En general a partir de datos etiquetados con calidad y el uso de técnicas correctas, las posibilidades de generar modelos de calidad basados en datos y aplicados a cualquier área del conocimiento son muy altas [5].

Algunas de las herramientas que pueden ser encontradas en este tipo de trabajos son:

Las redes LSTM las cuales son una arquitectura de red neuronal recurrente que ha demostrado ser efectiva para una variedad de tareas relacionadas con secuencias siendo capaces de retener y utilizar información a lo largo de secuencias largas, lo que las hace especialmente adecuadas para tareas que involucran contextos complejos y dependencias a largo plazo.

Por otro lado, BERT es un modelo de lenguaje basado en Transformers que ha tenido un gran impacto en el campo del procesamiento del lenguaje natural debido a su capacidad para capturar contextos bidireccionales y su alto rendimiento en una variedad de tareas de PLN.

2.1.3 Transformers

Los Transformers [6] tienen su origen en el 2017 y fueron descritos por primera vez por Google, siendo una de las clases más nuevas y potentes de modelos inventados actualmente, impulsando una ola de avances en aprendizaje automático que algunos han apodado como “La IA de Transformers “que inicialmente permitían realizar la traducción de un idioma a otro con la gran ventaja de poder entrenar al modelo en paralelo.

Esta tecnología presenta una arquitectura más compleja a comparación de las redes neuronales recurrentes pues a comparación de ellas en donde básicamente se cuenta con un codificador y un decodificador que funcionan de forma secuencial. Los Transformers cuentan un codificador encargado de asignar una secuencia de entrada de representaciones de símbolos a una secuencia de representaciones continuas para que posteriormente el decodificador genere una salida representada por una secuencia de símbolos por cada elemento. El uso de atención denominado como “multi-head” sucede en tres formas posibles:

- Cada posición en el decodificador para atender todas las posiciones en la secuencia de entrada. Esto imita el Mecanismos típicos de atención codificador-decodificador en modelos secuencia a secuencia.
- Cada posición en el codificador puede atender a todas las posiciones en la capa anterior del codificador.

- Cada posición en el decodificador atienda todas las posiciones en el decodificador hasta esa posición inclusive.

Desde esa perspectiva los Transformers han sido aplicados a problemas que involucran modalidades de entrada y salida distintas al texto y a la investigación de mecanismos locales de atención restringida para manejar eficientemente grandes entradas y salidas incluyendo imágenes, audio y vídeo. Básicamente los Transformers [7] han mejorado el estado de la técnica en diversas tareas de procesamiento del lenguaje natural tal como análisis de sentimientos, traducción y detección de misoginia en múltiples lenguajes.

2.2 Estado del arte

A continuación, se presenta una amplia descripción de trabajos relacionados al tema.

E. W. Pamungkas [8] habla de cómo, en los últimos años, el lenguaje de odio y en particular, el fenómeno del odio contra las mujeres, está aumentando exponencialmente en plataformas de redes sociales como Twitter y Facebook, a partir de ello los autores abordan las características más predictivas para distinguir entre contenido misógino y no misógino en redes sociales, además, de su relación con otros fenómenos de conductas de abuso y las cualidades en común para la detección de lenguaje de odio en cualquiera de sus categorías, logrando la construcción de un modelo de aprendizaje basado en LSTM y BERT, cuya experimentación consta de un dataset que comprende español, inglés e italiano. Dentro de sus conclusiones indica, que el modelo basado en BERT, es el mejor modelo en el experimento de configuración multilingüe, y determina así, que el discurso de odio hacia las mujeres es un fenómeno más relacionado con la misoginia que con el sexismo.

Partiendo de cifras concretas J. A. García-Díaz y sus colaboradores [9] brindan un panorama sobre la misoginia y su presencia en redes sociales particularmente en Twitter, nos mencionan que el 21% de las mujeres entre 18 y 29 años, denuncian haber sido agredidas en línea principalmente de forma sexual. Entre las actividades principales marcan la aplicación de tecnologías de análisis de sentimientos e informática social para la detección de mensajes misóginos en Twitter. Para la etapa de evaluación de las características, todos los experimentos fueron ejecutados en la plataforma WEKA versión 3.8.4 logrando así un modelo con una precisión de 85.175%.

Exponiendo que la misoginia en línea, considerada como un acoso, ha aumentado contra las mujeres árabes a diario A. Y. Muaad [10] afirma que la misoginia en el texto árabe ha recibido mucha atención en los últimos años debido a la discriminación racial y verbal, y, sobre todo, violencia contra las mujeres en las redes sociales. A partir de la construcción de un conjunto de datos denominado “Arabic Levantine Twitter Dataset for Misogynistic” y con el uso de distintos algoritmos de aprendizaje automático, como bosques aleatorios, SVM Linear, arboles de decisión, K vecinos más próximos, y también incluyendo AraBERT, el cual, es un modelo de idioma árabe, basado en la arquitectura BERT de Google, logran una precisión del 90,0 % y 89,0% para tareas binarias y multiclase, respectivamente.

Dentro del trabajo descrito por F. M. Plaza-Del-Arco [11] se encuentra un esfuerzo centrado en poder encontrar de forma específica la detección de comentarios contra mujeres e inmigrantes en redes sociales, específicamente en textos en español. Para lograr sus objetivos, se aplicaron diferentes técnicas de aprendizaje como NB, SVM, regresión logística, árbol de decisión y un

clasificador de votación por conjuntos. Todo esto utilizando una biblioteca de aprendizaje automático de software libre para la programación de Python, la mayor parte de su investigación trata sobre documentos escritos en inglés, argumentando que los recursos para el estudio de otros idiomas como el español eran insuficientes.

Con el auge que han tenido las redes sociales en los últimos años se ha permitido que, personas misóginas, xenófobas y homofóbicas, propaguen su discurso de odio, a fin de intimidar a cierto grupo de personas. J. A. García-Díaz [12] trabaja con un conjunto de datos sobre discursos de odio en España, que, a su vez, se enfocan en tres temas: misoginia, xenofobia y odio en general. A través del uso de tecnología transformer y rasgos lingüísticos se logra un modelo que, en su mejor versión, muestra una precisión de 90,4%, argumentando que el inglés, el español latinoamericano y el español europeo, siguen siendo un problema para los modelos de PNL, que a menudo, se basa en información léxica para sus tareas de clasificación.

G. Castillo-López [13] muestra un análisis exhaustivo de los modelos de detección del discurso de odio, comparando el estado del arte previo, tanto en la forma de entrenamiento de modelos como en los conjuntos de datos que existen y que han sido empleados como herramientas de entrenamiento en diferentes casos, concentrándose en diferentes variantes del español y empleando un conjunto de datos de Twitter crearon un modelo a partir de Beto, el cual a su vez está basado en BERT, para el español monolingüe funciona significativamente mejor que el BERT multilingüe para clasificar tweets como ofensivos, principalmente en dos dominios: misoginia y xenofobia, además de ello, resaltan que, las variantes de un lengua, debido por ejemplo, a su uso en diferentes países o culturas, afectan la actuación del odio en modelos de detección de voz.

Los Transformers han llegado a revolucionar la precisión de modelos predictivos, y pese a que si han sido aplicados en trabajos enfocados en conductas de odio y redes sociales actualmente la misoginia en contenido en español es un tema que se encuentra en etapas muy tempranas en cuanto a modelos que implementan esta tecnología. Además, en el presente trabajo se realiza un dataset que resulta del conjunto de tweets pero que también toma en cuenta contenido en Facebook añadiendo con el fin de enriquecer los datos de entrada y obtener resultados favorables en comparación a los trabajos existentes.

3 Detección de lenguaje misógino mediante transformers

Bajo el flujo que propone la metodología CRISP-DM (Figura 1) a continuación se describen los rasgos más relevantes para la realización de este proyecto y las actividades involucradas.

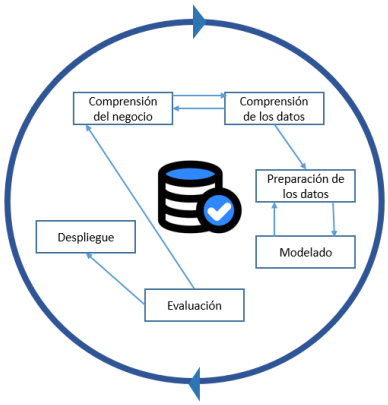


Figura 1. Flujo de metodología.

3.1 Dataset

En la literatura existen diversos conjuntos de datos empleados para la detección de lenguaje misógino en redes sociales, por ejemplo, [11] un conjunto de datos extraído de twitter para detectar misoginia y xenofobia. Por otro lado, MisoCorpus-2020 [9] es un corpus sobre misoginia que clasifica los comentarios en tres subconjuntos relativos a (1) violencia hacia mujeres relevantes, (2) mensajes de acoso a mujeres en español de España y español de Latinoamérica, y (3) rasgos generales relacionados con la misoginia. A pesar de la calidad de los conjuntos de datos antes mencionados, en este trabajo, se optó por desarrollar un conjunto de datos propio con el objetivo de garantizar que los datos sean relevantes para el problema que se intenta resolver, aprovechar los conocimientos de los expertos en la materia que permitan incorporar conocimientos matizados que podrían no estar presentes en los conjuntos de datos existentes, así como contribuir al aumento de datos en el contexto de detección de lenguaje misógina, ya que la cantidad de datos necesaria para los modelos de aprendizaje basados en Transformers es considerable.

3.1.1 Comprensión del negocio y de los datos

Para la construcción del dataset, se realiza la extracción de datos en Facebook empleando un Web Scraper con código en Python, que, a través de Url's que fueron identificadas por contener comentarios útiles y cuyos argumentos de búsqueda fueron nombres de mujeres que en la actualidad han sido objeto de comentarios violentos (Tabla 1), se realiza la extracción de los comentarios en su primer nivel de jerarquía de la publicación para formar parte del Dataset.

En cuanto a la extracción de datos en Twitter se utiliza código en Python que a partir del uso y manipulación de funciones programáticas brindadas por el API que esta red social ofrece, se desarrolló un módulo de recolección de tweets en volumen, a partir de la búsqueda de palabras relacionadas directamente a la misoginia, como se observa en la tabla 2.

3.1.2 Preparación de los datos

Con el fin de obtener un dataset que contenga elementos que en realidad sean capaces de sumar valor a la construcción del modelo objetivo se realizaron las siguientes tareas de pre procesamiento tal y como un trabajo que se apoya de NLP lo requiere:

- Eliminación de textos o comentarios repetidos.
- Eliminación de elementos que incluyen oraciones o contenido en idiomas ajenos al español.
- Eliminación de Textos u oraciones que se quedan truncados o terminan con '…'. (Caracteristica que generalmente fue detectada en Re tweets de Twitter y Replies de Facebook).
- Limpieza de etiquetas de usuario (Identificados con @) y atributos de Re tweets.

Tabla 2. Información extraída de Facebook.

FACEBOOK	
Nombre usado como argumento de búsqueda	**Numero de comentarios obtenidos**
Yalitza Aparicio	320
Tattis Beauty	110
Halle Bailey	266
Mon Laferte	124
Gloria Trevi	184
Bárbara de Regil	206
Karely Ruíz	157
Danna Paola	85
Rebeca Mendiola	26
Belinda Peregríni	39
Alondra Castro	45
Sol León	128
Adela Micha	142

Tabla 3. Información extraída de Twitter.

FACEBOOK	
Palabra/Frase buscada	**Palabra/Frase buscada**
Facilota	125
Golfa	1685
Maldita criada	12
Maldita zorra	288
Mantenida	7395
Piruja	1177
Putita	19535
Vieja interesada	25
Vieja inútil	758
Vieja Pendeja	787
Puta	0
Facilota	125

Una vez teniendo un dataset depurado se procedió a la etiqueta de sus elementos, en donde se asigna el valor de 0 para aquellos que no presentan contenido ofensivo y el valor de 1 en donde se detecta algún tipo de agresión misógina, tal como se muestra en la tabla 3.
Después de una primera revisión de los tweets etiquetados (3250 elementos) bajo la supervisión de un experto en el tema de conductas misóginas y su detección, se determinó que, pese a que algunos elementos contenían palabras que inicialmente fueron indicadas como clave para el etiquetado de datos, realmente carecían de un contexto semántico concreto que permita enriquecer al modelo planeado, por lo que se adicionan las siguientes acciones:

- Eliminación de elementos con menos de 50 caracteres.

- Enriquecimiento del Dataset con oraciones en donde se utilizan las palabras inicialmente identificadas, en donde se empleen sin un sentido misógino.

Tabla 3. Ejemplos de etiqueta manual de datos.

Texto	Valor de etiqueta
"Adiós a mi sueño de ver a Eiza González como Evelyn :(me encanta como actriz y aparte es hermosa"	0
"Zorra ya estoy muerto, perdón si te mentí adiós piruja."	1

Finalmente se obtiene un dataset compuesto de 2200 elementos los cuales se encuentran con una polaridad balanceada (50% con etiqueta 0 y 50% etiquetados con 1), en el cual se observan las siguientes características:

En la figura 2 se representa la longitud de caracteres en cada elemento del conjunto, observando así que se componen de textos, cuyo rango se encuentra entre los 50 y los 450 caracteres, percibiéndose mayor concurrencia en los primeros 150 caracteres, esto comprobando que los textos menores a 50 caracteres fueron depurados.

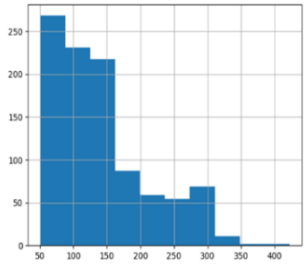


Figura 2. Longitud de palabras en los elementos del conjunto de datos.

En la figura 3 se destacan las palabras más frecuentes en el contenido del conjunto de datos siendo la palabra 'puta' la más encontrada con una frecuencia superior a 400 veces, palabras como: 'vieja', 'inútil', 'pendeja' y 'zorra' aparecen también en un alto número de elementos, es importante recalcar que pese a que estas palabras tienen alta frecuencia en el Dataset muchos elementos que las contienen las usan sin tener un significado ofensivo.

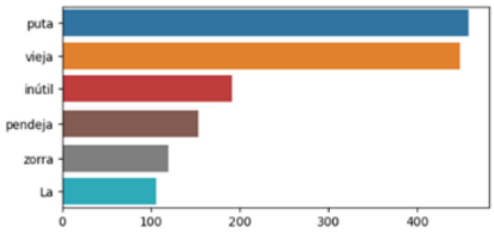


Figura 3. Frecuencia de las palabras con mayor presencia en el conjunto de datos.

3.2 Modelo Para la Detección de Lenguaje Misógino Basado en Transformers

3.2.1 Modelado

Con el Dataset previamente descrito, se procede con el entrenamiento para la obtención del modelo basado en Transformers, a continuación, se presentan los valores obtenidos, en la figura 4 se aprecia que, durante las tres iteraciones, se logra una perdida en el entrenamiento y validación, que ronda en cifras bastante cercanas entre una iteración y otra lo cual se refleja en la exactitud que en los tres casos es superior a un 75% siendo 80% el valor más alto.

Epoch	Training Loss	Validation Loss	Accuracy
1	0.513200	0.960224	0.802083
2	0.645100	0.826237	0.786458
3	0.414200	0.904688	0.802083

Figura 4. Resultados del entrenamiento del modelo basado en Transformers.

3.2.2 Evaluación

Para evaluar el modelo generado, es decir, para determinar qué tan preciso es el modelo para clasificar un comentario extraído de una red social como misógino o no misógino, se implementó la técnica de validación cruzada ya que ofrece resultados con un valor informativo más alto que métodos de validación regulares, además de que es recomendado para validar modelos de aprendizaje automático en un conjunto de datos limitado. El procedimiento de validación cruzada cuenta con un único parámetro llamado K que se refiere al número de grupos en los que se va a dividir una muestra de datos dada. Para este trabajo se empleó un valor de 10 para K, es decir, el conjunto de datos se dividió aleatoriamente en 10 partes, 9 de las cuales son utilizadas para el entrenamiento y una se emplea para las pruebas. Este proceso se repite 10 veces donde cada vez se utiliza una décima parte distinta para las pruebas. Los resultados de este proceso de validación se observan en la Tabla 4.

Tabla 4. Medición de la precisión, la recuperación y la medida F1
mediante validación cruzada de 10 veces.

Precisión	Recuperación	Medida F1
0.76	0.94	0.84

3.3 Resultados y despliegue

En la tabla 5 se muestra la interacción real con el modelo exponiéndolo a Tweets y comentarios que no pertenecen al conjunto de datos con el que fue entrenado, en ella podemos contrastar el valor esperado con el valor arrojado por el modelo, para finalmente, comparar la coincidencia de ellos.

Tabla 5. Prueba del modelo con elementos no incluidos en el conjunto de datos de entrenamiento.

Texto Evaluado	# caracteres	Valor esperado	Valor obtenido	Coincidencia de valores
"Eso piensa la zorra de tu madre de ti, pobrecita que le salió tan retrasado el niño."	84	1	1	SI
"Nuestro país y sus pinches leyes que para algunos privilegiados hasta las sentencias se reducen, por eso más personas sin valores, normas hacen lo que se les pega la gana. Bravo"	177	0	0	SI
"Ese es el diario vivir de la Colombia criada a punta de violencias de toda clase desde hace muchos años con personajes que ya saben quiénes Fueron porque ahora hay otro actor más grave: La Maldita corrupción rampante e impunidad increíble en todos los ámbitos del país."	269	0	1	NO
"A mí que chucha que puedas viajar piruja, por qué las ganas de querer compararnos entre mujeres. Suficiente tenemos con los hombres quitándonos espacios para que ahora nos toque lidiar con babosas cómo está"	206	1	1	SI
"Que pocos años le dieron ahora al pareser ya salió No se puede creer que las autoridades dejen pasar por alto una violación Por eso estamos como estamos"	151	0	0	SI

4 Conclusiones y trabajo a futuro

Un trabajo de esta naturaleza debe contar con una selección de datos bien fundamentada, es decir, entre más cuidadosa sea la selección y clasificación de los datos que usan durante el entrenamiento del modelo, mejores resultados se obtienen en un entorno de despliegue real.

También cabe resaltar la importancia del acompañamiento de un especialista en el área, que en el proceso resulta fundamental, especialmente hablando en el caso de análisis de lenguaje natural, ya que, en la etapa de etiquetar a cada uno de los elementos que conforman al conjunto de datos, llegan a surgir dudas que, de no ser resueltas, podrían afectar al proceso y entrenamiento del modelo. Como se pudo observar en los ejemplos, el uso del modelo obtenido, sí es capaz de identificar el contenido (objetivo) en texto que es ingresado como entrada, pero de aquellos que formaron parte del margen de error, se observa que su corrección podría influir en el hecho de poder incluir dentro del dataset elementos que, a pesar de contener palabras de connotación positiva, llevan implícito un mensaje despectivo, tal como se puede observar en la siguiente oración: *"No cabe duda de que eres una princesa hermosa pero una piruja en la cama y una perra en acción"*

En este tipo de oraciones se da el caso de una mala clasificación, porque son asociadas directamente con las palabras claves o forman parte de la descripción del sujeto, en este caso "princesa, hermosa y puta" pasan a ser los adjetivos calificativos, pero con el hecho de tener un halago entre ellas su clasificación no logra detectar la parte despectiva de forma óptima, es por ello que se consideran las siguientes recomendaciones para la continuidad y optimización de este proyecto:

- La adición de elementos que contengan misoginia con un tono de sutileza es decir que no necesariamente incluyan palabras altisonantes para el español.
- Enriquecer el conjunto de datos agregando elementos que contengan tanto contenido halagador como elementos misóginos como el siguiente ejemplo: *"No cabe duda de que eres una princesa hermosa pero una piruja en la cama".*

- Incluir una nueva clase que permita la identificación de la variante de español o el país de origen, podría ser interesante y enriquecedor para un modelo que pueda considerar el uso de una palabra en diferentes contextos.

Agradecimientos

Agradecemos al Tecnológico Nacional de México (TecNM)/Instituto Tecnológico Superior de Teziutlán por apoyar este trabajo. También al sistema DIF del municipio de Chignautla, Puebla por el acompañamiento y asesoramiento de especialistas en el área psicológica. Esta investigación también fue patrocinada por el Consejo Nacional de Humanidades, Ciencias y Tecnologías de México (CONAHCYT).

Referencias

[1] INEGI, "MÓDULO SOBRE CIBERACOSO 2020," 2021.

[2] D. Cerva Cerna and D. Cerva Cerna, "La protesta feminista en México. La misoginia en el discurso institucional y en las redes sociodigitales," Rev Mex Cienc Polit Soc, vol. 65, no. 240, pp. 177–205, Aug. 2020, doi: 10.22201/FCPYS.2448492XE.2020.240.76434.

[3] "Misoginia y violencia lingüística en las redes sociales - El Sol de México | Noticias, Deportes, Gossip, Columnas." https://www.elsoldemexico.com.mx/analisis/misoginia-y-violencia-linguistica-en-las-redes-sociales-4549306.html (accessed Jun. 28, 2023).

[4] "¿Qué es el aprendizaje automático? | Oracle México." https://www.oracle.com/mx/artificial-intelligence/machine-learning/what-is-machine-learning/ (accessed Jun. 28, 2023).

[5] R. Ramos Pollán, "Perspectivas y retos de las técnicas de inteligencia artificial en el ámbito de las ciencias sociales y de la comunicación," Anuario Electrónico de Estudios en Comunicación Social "Disertaciones," vol. 13, no. 1, pp. 21–34, Jan. 2020, doi: 10.12804/revistas.urosario.edu.co/disertaciones/a.7774.

[6] A. Vaswani et al., "Attention Is All You Need," Jun. 2017, [Online]. Available: http://arxiv.org/abs/1706.03762.

[7] Y. Qiang, D. Pan, C. Li, X. Li, R. Jang, and D. Zhu, "AttCAT: Explaining Transformers via Attentive Class Activation Tokens", Accessed: Aug. 28, 2023. [Online]. Available: https://github.com/qiangyao1988/AttCAT.

[8] E. W. Pamungkas, V. Basile, and V. Patti, "Misogyny Detection in Twitter: a Multilingual and Cross-Domain Study," Inf Process Manag, vol. 57, no. 6, Nov. 2020, doi: 10.1016/j.ipm.2020.102360.

[9] J. A. García-Díaz, M. Cánovas-García, R. Colomo-Palacios, and R. Valencia-García, "Detecting misogyny in Spanish tweets. An approach based on linguistics features and word embeddings," Future Generation Computer Systems, vol. 114, pp. 506–518, Jan. 2021, doi: 10.1016/j.future.2020.08.032.

[10] A. Y. Muaad, H. J. Davanagere, M. A. Al-antari, J. V. B. Benifa, and C. Chola, “AI-Based Misogyny Detection from Arabic Levantine Twitter Tweets,” MDPI AG, Mar. 2022, p. 15. doi: 10.3390/ioca2021-10880.

[11] F. M. Plaza-Del-Arco, M. D. Molina-González, L. A. Ureña-López, and M. T. Martín-Valdivia, “Detecting Misogyny and Xenophobia in Spanish Tweets Using Language Technologies,” ACM Trans Internet Technol, vol. 20, no. 2, May 2020, doi: 10.1145/3369869.

[12] J. A. García-Díaz, S. M. Jiménez-Zafra, M. A. García-Cumbreras, and R. Valencia-García, “Evaluating feature combination strategies for hate-speech detection in Spanish using linguistic features and transformers,” Complex and Intelligent Systems, vol. 9, no. 3, pp. 2893–2914, Jun. 2023, doi: 10.1007/s40747-022-00693-x.

[13] G. Castillo-López, A. Riabi, and D. Seddah, “Analyzing Zero-Shot Transfer Scenarios across Spanish variants for Hate Speech Detection,” 2023. [Online]. Available: https://www.ethnologue.com/ethnoblog.

Interfaz de un sistema de reconocimiento de colores (RGB) basado en la red neuronal ADALINE

Interface of a color recognition system (RGB) based on the ADALINE neural network.

Mauro Alberto López Muñoz, Guillermo Colorado Jiménez, Nancy Montalvo Montalvo, Mauricio Torres González, Cesar Augusto Arriaga Arriaga
Benemérita Universidad Autónoma de Puebla, Av. San Claudio esq. 18 Sur edif. FCE1, Col. San Manuel, Ciudad Universitaria. C.P. 72570. Puebla, Puebla, México.
mauro.lopez@alumno.buap.mx, guillermo.coloradojim@alumno.buap.mx, nancy.montalvo@alumno.buap.mx, mauricio.torresg@alumno.buap.mx, cesarau.arriaga@correo.buap.mx.

Abstract

The rise of artificial intelligence (AI), and particularly the use of neural networks for its development, has made it possible for machines to be trained to perform specific tasks by processing data and recognizing patterns in them. The use of graphical interfaces allows users to interact with the machines in a friendly manner and fulfill a given task. The work aims to use an ADALINE neural network to train a system that recognizes objects according to their color using a graphical interface for their physical implementation. The training and validation of the system was carried out in order to demonstrate its efficiency, as well as the opportunity points of a prototype of this nature. The results indicate that the project developed is able to achieve a high degree of reliability with a training from 40 samples making it effective to detect 2 different classes. It is concluded that the work is highly scalable in different fields, in addition that, depending on the complexity of the use case, its precision can be fine-tuned by doing more rigorous training.

Resumen

El auge de la inteligencia artificial (IA), y particularmente el uso de redes neuronales para su desarrollo, ha hecho posible que las máquinas puedan ser entrenadas para realizar tareas específicas procesando datos y reconociendo patrones en ellos. El uso de las interfaces gráficas permite que los usuarios interactúen con las máquinas de manera amigable y cumplan una tarea dada. El trabajo tiene como objetivo usar una red neuronal ADALINE para entrenar a un sistema que reconozca objetos según su color usando una interfaz gráfica para su implementación física. Se realizó el entrenamiento y validación del sistema con el fin de demostrar su eficiencia, así como los puntos de oportunidad de un prototipo de esta naturaleza. Los resultados indican que el proyecto desarrollado es capaz de alcanzar un alto grado de fiabilidad con un entrenamiento a partir de 40 muestras haciéndolo eficaz para detectar 2 clases diferentes. Se concluye que el trabajo es altamente escalable en diferentes campos, además que, dependiendo de la complejidad del caso de uso, se puede afinar su precisión haciendo un entrenamiento más riguroso.

Keywords and phrases: Red Neuronal ADALINE, Interfaz gráfica, LabVIEW, RGB.

1 Introducción

Las redes neuronales artificiales surgen como un enigma asociado al interés que el hombre ha tenido por conocer el funcionamiento de su propia naturaleza, que caracteriza a la especie humana del resto de los seres vivos, las cuales han tenido gran impacto en el campo computacional, ya que tienen la capacidad de aprender a partir de modelos y patrones que están presentes en la información a base de entrenamiento, capaces de resolver problemas complejos y se han convertido en la base fundamental de la Inteligencia Artificial.

Las interfaces gráficas de usuario son los elementos gráficos que nos ayudan a comunicarnos con un sistema o estructura [1] y sirven a los usuarios como un mediador para controlar de manera más sencilla los procesos o tareas computacionales, permitiendo un entorno más amigable e intuitivo para poder realizar cualquier aplicación en diferentes campos de la vida diaria, además de encontrarse en constante desarrollo. Gracias a ellas es posible conectar el hardware con las computadoras, procesar la información, establecer parámetros de control y de esta manera tomar decisiones oportunas o automatizar procesos con el fin de ahorrar tiempo y esfuerzo humano.

En el presente trabajo se muestra el desarrollo de una interfaz gráfica con un sistema de reconocimiento de colores en la gama RGB basado en una red neuronal, comenzando con su desarrollo, pasando por las etapas de entrenamiento basado en la información que el usuario provee, la validación para asegurar que el sistema detecta y clasifica de manera adecuada, así como el diseño de la interfaz para concluir con su implementación.

2 Marco teórico y estado del arte

2.1 Red neuronal ADALINE

La red ADALINE (por su acrónimo en inglés **ADA**ptative **L**inear **NE**uron) es uno de los modelos clásicos de las redes neuronales, creada por Widrow y Hoff en 1959 [2]. Esta neurona se asemeja al perceptrón simple, la diferencia radica en la utilización de una función de transferencia en lugar de usar la función signo, su salida es entonces una función lineal de las entradas (ponderadas con los pesos sinápticos), como se muestra en la ecuación 1.

$$y = \sum_{j=1}^{1N} w_j * x_j - 0 \qquad (1)$$

Siendo N, un número mayor que 1, corresponde a los elementos de entrada que está asociado con un peso ajustable de número real. Al calcular la suma de los elementos de entrada ponderados más un sesgo para producir una salida lineal y luego alimentando la salida lineal a una función de

activación para producir una salida. Si se considera una entrada adicional con un valor X_{N+1}=-1 cuyo peso sináptico es $W_{N+1} = 0$, entonces se puede decir que:

$$y = \sum_{j=1}^{N} w_j \, x_j \tag{2}$$

De esta forma se puede generalizar la ecuación 2, donde se puede tratar al 0, como un peso sináptico adicional.

En general la red ADALINE pretende implementar la correspondencia dada por las entradas y las salidas de un sistema utilizando un conjunto finito de relaciones entre ellas. Consideremos que se dispone de $\boldsymbol{p}$ patrones como entrada $x^1, x^2, \dots, x^p$, y sus correspondientes salidas $z^1, z^2, \dots, z^p$, con el objetivo de determinar los pesos sinápticos que hagan que las salidas de la red sean más cercanas a las salidas deseadas para el conjunto dado de patrones de entrenamiento, es decir, determinar los pesos sinápticos de manera que se minimice la función de error cuadrático dado por la ecuación 3:

$$E = \frac{1}{2}\sum_{k=1}^{p}(z^k - y(k))^2 \; = \; \frac{1}{2}\sum_{j=1}^{p}(z^k - \sum_{j=1}^{N+1} w_j \, (k){x^k}_j)^2 \tag{3}$$

Con el método de descenso del gradiente es posible minimizar el error, si en la iteración k se introduce el patrón de entrenamiento x^k, cuya salida deseada es z^k y los pesos sinápticos son $w_j(k)$, donde $j = 1,2, \dots, N$, entonces la modificación se expresa como ecuación 4:

$$w_j(k+1) = \; w_j(k) + \Delta w_j(k) \tag{4}$$

Donde el parámetro η controla la longitud del paso opuesta al gradiente. Cuanto mayor sea η de igual manera será la cantidad que se modificarán para los pesos sinápticos, es por ello por lo que el parámetro debe de ser un valor pequeño para evitar dar pasos demasiado largos, es decir, que nos lleven a soluciones peores, debido que el método del gradiente solamente garantiza el decrecimiento de la función de error si nos desplazamos en la dirección opuesta, pero en un entorno suficientemente pequeño.
Esta red neuronal artificial de una sola capa se utiliza principalmente para problemas de clasificación binaria y regresión lineal, así como reconocimiento de patrones, predicción financiera, control de procesos industriales y análisis de datos, en general es una herramienta muy útil en la inteligencia artificial y el aprendizaje automático, y se utiliza en una amplia gama de aplicaciones en varios campos, desde la visión artificial hasta la predicción financiera y el control de procesos industriales.

2.2 LabVIEW

El entorno de programación LabVIEW está basado en elementos gráficos, así como bloques que simplifican el proceso de codificación y permiten al usuario enfocarse en su problema de ingeniería. Además, está diseñado para una fácil integración con el hardware, lo que hace posible construir sistemas de adquisición de datos y su posterior gestión para obtener información clara y precisa mediante la utilización de tablas, imágenes, así como otros elementos gráficos que permiten al usuario final una interacción amigable y fluida [3].

El software cuenta con una amplia variedad de bloques, además de ofrecer la posibilidad de crear estructuras propias completamente personalizadas, es decir que es posible diseñar bloques con entradas y salidas que cumplan una función específica y que puede integrarse con las herramientas incorporadas del sistema. Por otro lado, se pueden integrar diferentes diseños desarrollados en LabVIEW para crear un proyecto más grande y tan fácil o complejo como se desee, asegurando siempre la simplicidad de uso para el usuario final.

2.3 Trabajos relacionados

La red neuronal ADALINE ha sido ocupada ampliamente en el sector educativo e industrial, y particularmente para el presente escrito, se ha utilizado para modelos de reconocimiento o clasificación, así como modelos predictivos. A continuación, se recopila un estudio acerca del uso de redes neuronales en conjunto con otras herramientas para lograr proyectos relacionados a lo que compete este artículo.

En [4], en 2022 se desarrolló un sistema para detectar patrones en piezas de artesanías, partiendo de una imagen digital que se procesa para pasar del espacio de color RGB al espacio CIE L * a * b. Se utilizó un algoritmo para agrupar y analizar los diferentes matices de las piezas, para finalmente usar una plataforma de estadística inteligente para identificar los diferentes colores de esmalte de 2 tipos de artesanías de la dinastía Ming con el cual se estudió el índice de cromaticidad de cada muestra. Por otro lado, en [5], como parte del concurso internacional de innovación de UAV de 2019, se estudia la imagen de luz de dirección de color en tiempo real recopilada por la cámara del UAV y propone un modelo de reconocimiento automático de color de UAV basado en visión artificial, mientras tanto, en [6] se diseñó un sistema impulsado por una red neuronal convolucional para separar hasta 5 categorías basadas en el color y así determinar la madurez de aguacates. Además, en [7], se exponen diferentes tipos de redes neuronales pasando por el Perceptrón, hasta el MADALINE, abarcando proyectos prácticos que van desde simulación y graficado hasta aplicaciones prácticas en las que se incluye la detección de colores de 2 clases de objetos para su posterior clasificación.

3 Arquitectura de la aplicación

El desarrollo del presente trabajo se basa en dos bloques principales, la codificación de la red neuronal, que es la parte encargada de hacer el proceso de entrenamiento y el otro bloque es la validación de las muestras proporcionadas por el usuario, para lo cual se desarrolló el programa siguiendo el diagrama que se encuentra en la Figura 1, cuya base se centra en el funcionamiento de la red neuronal ADALINE, en la cual se declara el vector {x} de muestras de entrenamiento, posteriormente se asocia la salida deseada {d} para cada muestreo, después de que se tienen definidos estos parámetros, se inicializa el vector {w} con pequeños valores aleatorios, luego se especifica el valor de la tasa de aprendizaje {n} y de precisión {e}, para iniciar con el conteo de épocas, cuyo valor es el número de muestras que se va a repetir el ejercicio de muestreo y así ajustar el valor de los pesos {w}. Para determinar que la red está completamente entrenada, se determinará cuando el valor del error cuadrático medio entre dos épocas sucesivas sea menor que el valor de la precisión {e}. Una vez que se tienen en consideración estos dos criterios, se detendrá el número de

épocas que pasaron para poder obtener los pesos deseados para tener la clasificación adecuada de los objetos. A su vez se continuará con el segundo bloque que es la validación de los pesos obtenidos con la toma de muestras aleatorias para la clasificación de estas [7].

3.1 Estructura de la interfaz gráfica con Labview

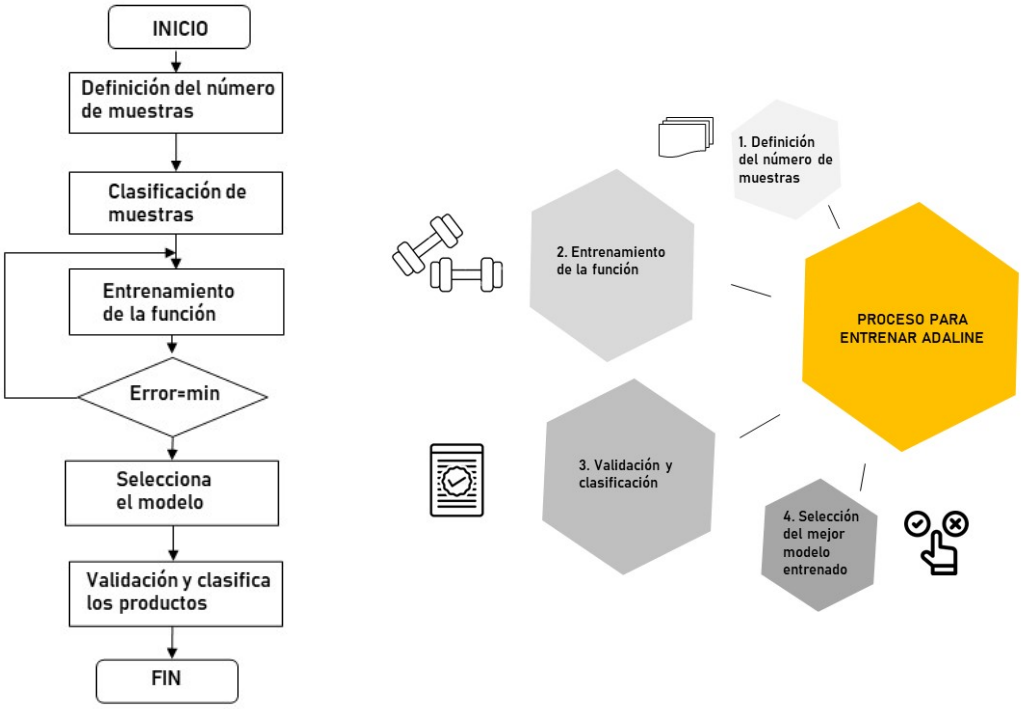


Figura 1. Diagrama de flujo y de bloques del entrenamiento y selección de modelo de ADALINE.

El código del ADALINE se grabó en una ESP32 de la empresa Espressif Systems, que pertenece a la categoría de system on a chip (SoC, por sus siglas en inglés), la comunicación se realizó a través del modo serial con el protocolo RS232, con lo que se obtuvieron 2 puntos, la computadora se configuró para tener el rol de maestro y el microcontrolador el rol de esclavo. Este último sigue las indicaciones que el usuario le manda a través de la interfaz gráfica (HMI, por sus siglas en inglés) como lo muestra la figura 2, de esta forma el usuario tiene el control total del proceso de recolección de muestra pues en la interfaz desarrollada en el software

LabVIEW se puede definir qué tipo de muestra se está capturando. Adicionalmente la interfaz cuenta con un botón de cálculo de pesos que se realizó a través del entrenamiento de la toma de muestras de los objetos.

La comunicación del microcontrolador con el sensor de color RGB TCS34725 es a través de Inter-Integrated Circuit (I2C, por sus siglas en inglés) que es un protocolo de comunicación serial síncrono de dos cables, una para la transmisión de datos (SDA, por sus siglas en inglés) y otra para la señal de reloj (SCL, por sus siglas en inglés), y permite la comunicación de varios dispositivos en el mismo bus de comunicación.

Figura 2. Diagrama de la interfaz Humano Máquina.

En el panel frontal que se presenta en la figura 3 se observan los botones necesarios para la toma de muestras, identificando cada botón dedicado a los dos tipos de muestra que se tomarán, empezando por el botón de la muestra 1, repitiendo el proceso 20 veces, el progreso se observa en el indicador llamado conteo de muestra, una vez que se completa el número máximo de repeticiones de la muestra 1 se repite el proceso para la muestra 2 con el botón dedicado a esa función, una vez que se alcanza el valor de muestras capturadas, se presiona el botón que calculará los pesos de las muestras y finalmente para identificar la muestra se presiona un botón llamado resultado, que mostrará lo obtenido en un indicador de texto.

Para la validación del experimento se tomaron dos tipos de muestras de frutas, los cuales son naranjas y manzanas verdes, con el fin de que el sensor pueda identificar de mejor manera cada uno de los colores.

Con el fin de tener un reconocimiento de colores con una mayor precisión, se determinó partir de 20 muestras para cada clase, debido a que mientras más grande sea el campo de muestreo, el error irá disminuyendo.

Posteriormente para realizar el experimento se muestrearon una por una las frutas, girando el sensor en cada parte de la muestra con el fin de que capture el color en distintas zonas de la fruta, con el objetivo de comprobar el experimento y obtener un resultado en vectores de valores de la gamma RGB, es decir, hay un valor para cada color que indica el espectro del mismo y su intensidad en verde, rojo y azul, obteniendo un resultado adecuado sin importar la zona de la fruta, como se muestra en la figura 4.

Una vez que se tienen las muestras y los colores de cada fruta cargados en el sistema de la ESP32 se procede a iniciar el cálculo de los pesos con su respectivo botón en el panel frontal, en la figura 5a, se observa el sistema general del experimento ya con las muestras tomadas de los datos, en la figura 5b se observan los resultados en mostrados por el panel frontal, comprobando así el funcionamiento del sistema.

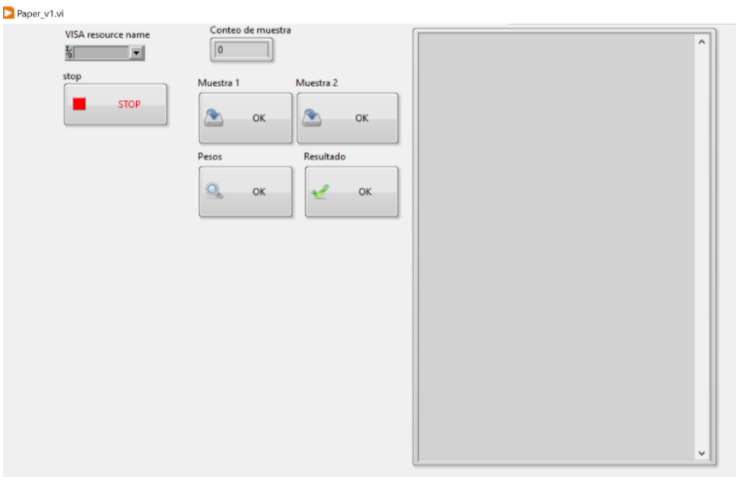


Figura 3. Panel virtual del sistema.

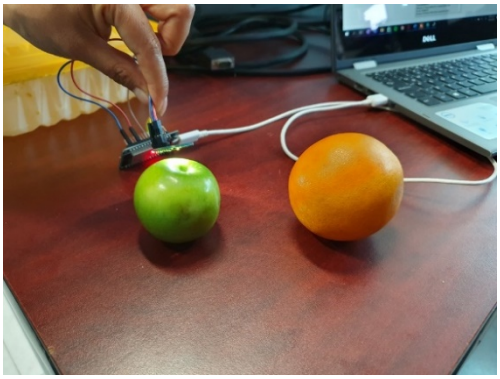


Figura 4. Muestreo del experimento.

4 Resultados

La base de entrenamiento del modelo se basa en la repetición en la toma de muestras, ya que mientras más muestras se escaneen, el error irá disminuyendo a la hora de mostrar resultados, esto con el fin de que el ADALINE distinga de una mejor manera las muestras con las que se le entrenó y sea capaz de distinguir otros elementos de la misma familia, con características no tan marcadas como con las que se entrenaron, los resultados del número de muestras, épocas, erro y los pesos actuales de cada una se observan en la tabla 1.

En las gráficas se observa el comportamiento de los distintos resultados en comparación con el número de toma de muestras, como se observa en la figura número 6a que es la comparativa de las épocas con respecto al número de muestras, en esta se observa que el mayor número de épocas fue en la toma 40 y después fue descendiendo el número de épocas que necesitaba para poder encontrar los pesos adecuados. En la figura número 6b se tiene el error que se obtuvo en el número de muestras tomadas, esta se basa con el error cuadrático y la precisión que se quiere tener en la red neuronal, la cual en este caso fue de 0.00001, por ello este es el valor más grande que puede tener de error el sistema, como lo fue en la toma 30 y 40, estos tuvieron un error muy grande que

cumplieron el mínimo para que el sistema no repitiera el cálculo de los pesos, en cambio los errores más pequeños fueron de la toma de muestra 20 y 60.

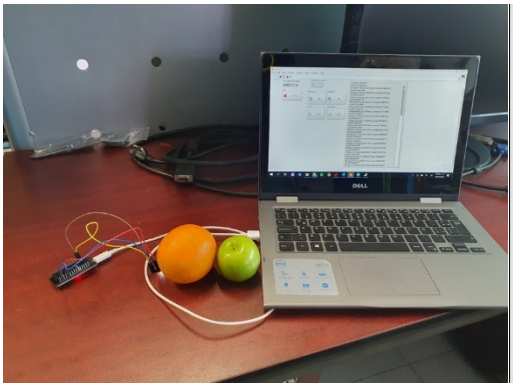


Figura 5a. Funcionamiento del experimento.

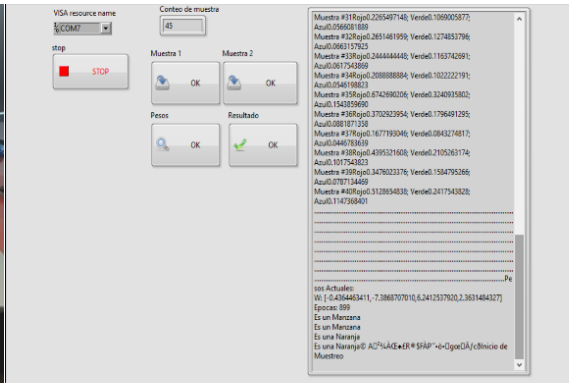


Figura 5b. Funcionamiento del panel virtual del experimento.

Tabla 4. Comparativa de los distintos resultados en el muestreo.

No. De Muestras	Épocas	Error	Pesos Actuales			
			W1	W2	W3	W4
20	1295	9.96748E-06	-0.874521853	4.8481355078	-11.8420127190	2.3127854474
30	3244	9.99868E-06	-1.2080699456	3.5534646744	-14.6111241261	0.5226547946
40	5760	9.99868E-06	-0.1074468722	10.4738618234	-12.9891514540	-1.2593114402
50	2734	9.99123E-06	0.7114327193	14.3168262252	-16.9625437264	1.6180297956
60	1467	9.96143E-06	-0.4667572549	7.3668340629	-14.0771307315	-0.0430537698
70	1220	9.98378E-06	0.1853102208	5.1457973722	-6.5564696768	1.9588183806

En la figura número 6b se ve los cuatro pesos necesarios para poder obtener el valor deseado con respecto al número de tomas de muestras, son cuatro pesos por que el sensor RGB que se utilizó que corresponden a él Red, Green, Blue y Clear, estos valores se obtienen en cada toma de muestra. Como se observa en la gráfica de la figura 6b los pesos permanecen en un margen, ya que esto se debe a que a que todos tiene que cumplir con el criterio de obtener el valor deseado cuando se ejecute la clasificación del objeto.

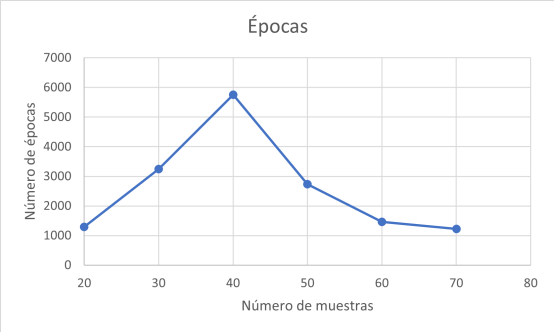


Figura 6a. Grafica de las épocas contra No. de muestras.

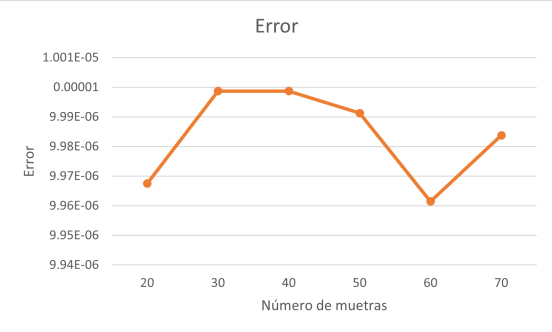


Figura 6b. Grafica del error contra No. de muestras.

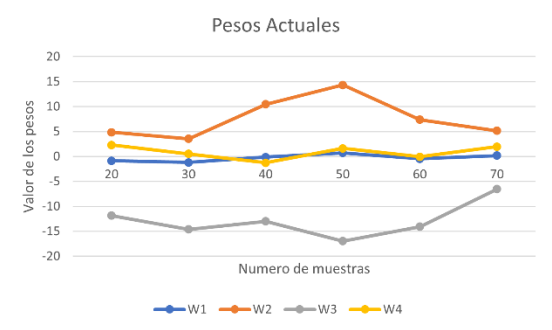


Figura 6c. Grafica de los pesos actuales contra No. de muestras.

5 Conclusiones y trabajo a futuro

El sistema desarrollado fue capaz de identificar 2 clases, en este caso una naranja y una manzana verde, con lo que, tras un entrenamiento de 40 muestras totales, se realizó el proceso de validación con 33 pruebas, obteniendo sólo dos errores en el proceso, lo que indica que para obtener el mayor grado de precisión se requiere de un entrenamiento con más muestras. Esto se pudo concluir ya que se tomaron diferentes cantidades de muestras empezando con 20, en donde no se pudo obtener el error cuadrático deseado llevando a tener resultados incorrectos en la implementación, por lo tanto, se tomó un mayor número de muestras, llegando hasta 100, con lo que se obtuvo una mayor precisión, sin embargo, el tiempo para calcular los pesos se elevó. Por lo tanto, para lograr una implementación óptima es importante considerar las muestras del entrenamiento y el tiempo de ejecución de la red. Es importante mencionar que a pesar de que en ocasiones se requiere de un entrenamiento aparentemente tardado, una vez finalizado este proceso el sistema conservará su alto grado de reconocimiento en cuanto a las clases agregadas. Finalmente, el proyecto desarrollado resultó ser una opción fiable, accesible y de bajo costo para clasificar 2 tipos de elementos.

Uno de los trabajos a futuro que representa un verdadero desafío para la continuación del trabajo es mejorar el algoritmo que permita entrenar un modelo que sea capaz de clasificar elementos de distintos matices dentro de una misma tonalidad de color, es decir trabajar en un rango de colores más cercanos que permita distinguir entre elementos similares, que puede aplicarse para la recolección y clasificación de frutos y vegetales, abriendo una rama de desarrollo si se desea detectar los niveles de maduración de una misma especie.

Otra área de oportunidad es el Desarrollo de una plataforma virtual educativa, la propuesta queda abierta a introducir más elementos clasificatorios para el desarrollo de una herramienta educativa que coadyuve al desarrollo y aprendizaje de las redes neuronales a estudiantes de ingeniería en el entrenamiento de modelos para la generación de redes neuronales más eficientes.

Agradecimientos

Los autores de este artículo agradecen al Consejo Nacional de Humanidades Ciencia y Tecnología (CONAHCYT) y a la Benemérita Universidad Autónoma de Puebla por el apoyo otorgado en la realización de este trabajo.

Referencias

[1] L. L. González, "Diseño de una interfaz gráfica de usuario para publicaciones digitales". México: Publicaciones Digitales, DGSCA, UNAM., 2004.

[2] P. I. V. e. I. M. GALVAN, "Redes neuronales artificiales: un enfoque práctico". Prentice Hall, 2004.

[3] "National Instruments". [Online]. Available: https://www.ni.com/es-mx/support/downloads/software-products/download.labview.html#477380. [Accessed: 20/Abril/2023].

[4] Z. Kuang, "Application of Color Recognition and Pattern Analysis of Handicrafts in Intelligent Statistical Platform". International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, 2022.

[5] C. Z. Q. G. a. F. W. G. Liu, "Automatic Color Recognition Technology of UAV Based on Machine Vision". International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), Beijing, 2019.

[6] J. E. C. d. l. C. a. O. J. V. Ramirez, "Convolutional neural networks for the Hass avocado classification using LabVIEW in an agro-industrial plant,". IEEE XXVII International Conference on Electronics, Electrical Engineering and Computing (INTERCON), Lima, 2020.

[7] I. Nunes, D. Hernane Spatti, R. Andrade Flauzino, L. H. Bartocci Liboni y S. F. Reis Alves, "Artificial Neural Networks". Switzerland: Springer, 2017.

[8] R. D. a. I. Dogaru, "CONV - ELM: Binary Weights Convolutional Neural Network Simulator based on Keras/Tensorflow, for Low Complexity Implementations". 6th International Symposium on Electrical and Electronics Engineering (ISEEE), Galati, 2019.

[9] G. M. B. a. A. B. V. Muneer, "Optimal Machine Learning based Controller for Shunt Active Power Filter by Auto Machine Learning". IEEE Journal of Emerging and Selected Topics in Power Electronics, 2018.

[10] Y. W. y. K. Z. Xiaoqin Zeng, "Computation of Adalines sensitivity to weight pertubation". IEEE, vol. 17, nº 2, pp. 515-517, 2006.

Control neuronal adaptativo de un sistema de primer orden con incertidumbres

Neural adaptive control of a first-order system with uncertainties

Valentín García Cervantes[1], Amparo Dora Palomino Merino[1],
Juan Escareno[2], María Aurora Diozcora Vargas Treviño[1]

[1]Benemérita Universidad Autónoma de Puebla. Puebla, Puebla, Mexico.

[2]XLIM Research Institute UMR CNRS 7252, Limoges University, Limoges, France.

valentin.garciac@alumno.buap.mx, amparo.palomino@correo.buap.mx,
juan.escareno-castro@unilim.fr, aurora.vargas@correo.buap.mx.

Abstract

The development of autonomous systems is a quite important topic nowadays, and its relevance continues to grow in various fields of society. However, the autonomy of these new systems is limited due high computational cost and times. In this article, we propose an Adaptive Neural Control that utilizes a radial basis function neural network. These types of networks require fewer neurons in the hidden layer compared to other artificial neural networks, thus reducing the complexity of training and computational processing, resulting in energy efficiency. Utilizing the MATLAB - Simulink simulation software, the Adaptive Neural Control is implemented on a first order dynamic system with the presence of uncertainties. Finally, a comparative analysis of the proposed control with a classic PI Controller is conducted for cases involving constant disturbance, time-varying disturbance, and state-dependent disturbance.

Resumen

El desarrollo de sistemas autónomos es un tópico bastante importante en la actualidad, su relevancia continúa creciendo en diversos ámbitos de la sociedad. Sin embargo, la autonomía de los nuevos sistemas se ve limitada debido a los altos costos y tiempos computacionales. En este artículo se propone un control neuronal adaptativo que utiliza una red neuronal de tipo función base radial, este tipo de redes requieren menos neuronas en la capa oculta, comparadas con otras redes neuronales artificiales, lo que reduce la complejidad de entrenamiento y de procesamiento computacional, obteniendo por ende una eficiencia energética. Utilizando el software de simulación MATLAB – Simulink, el control neuronal adaptativo es implementado sobre un sistema dinámico de primer orden con presencia de incertidumbres. Finalmente, se realiza un análisis comparativo del control propuesto con un controlador PI clásico para los casos de una perturbación constante, perturbación variante en el tiempo y perturbación dependiente del estado.

Keywords and phrases: Redes Neuronales, Control Neuronal Adaptativo, Función Base Radial, PI Clásico.

---

1 Introducción

Actualmente las aplicaciones de sistemas autónomos se han multiplicado exponencialmente en diversos ámbitos sociales. Sin embargo, su autonomía se ve limitada por los altos costos y tiempos computacionales por lo que actualmente se buscan nuevas formas de conseguir una eficiencia energética. En este sentido, los algoritmos de control han evolucionado a fin de realizar tareas más complejas mostrando una autonomía elevada, en donde los sistemas de control neuronal se distinguen por la plasticidad y desempeño frente a condiciones inesperadas.

Una forma de ver a una red neuronal artificial (RNA) es como un algoritmo que a su vez se compone de diferentes algoritmos, que realizan cálculos locales más pequeños a medida que los datos se propagan a través de él [1]. Existen diferentes modelos de RNA que permiten solucionar problemas difíciles de resolver para algoritmos de control convencionales. Las redes neuronales tienen la capacidad de aprender, por lo que, si en un sistema conocemos la entrada, el valor de salida deseado y el valor de salida actual, un controlador integrado por una red neuronal es capaz de modificar sus parámetros aprendiendo la dinámica de la planta hasta conseguir un sistema confiable.

Las RNA se han aplicado ampliamente al control adaptativo de sistemas no lineales, su capacidad de aprendizaje, junto con su capacidad de capturar relaciones no lineales, les permite ajustar y optimizar el comportamiento del sistema de control para lograr una operación más eficiente desde el punto de vista energético [2, 3]. En algunos casos, el entrenamiento inicial de la red neuronal puede requerir una cantidad significativa de energía y recursos computacionales, sin embargo, una vez entrenada, la capacidad de las redes neuronales para adaptarse, aprender y optimizar su comportamiento puede contribuir significativamente a mejor la eficiencia energética en diversas aplicaciones de control [4].

En este trabajo se realiza la implementación mediante simulación de un control neuronal adaptativo, el cual utiliza una red neuronal del tipo función base radial (RBF) sobre un sistema dinámico de primer orden en presencia de diferentes tipos de incertidumbres; perturbación contante, perturbación variante en el tiempo y perturbación dependiente del estado.

2 Marco teórico y estado del arte

Las redes neuronales de función base radial (RBF por sus siglas en inglés) han demostrado tener una buena capacidad de aproximación no lineal de manera rápida [5]. Las funciones de activación en una RBF son Gaussianas, estas permiten modelar relaciones no lineales complejas en los datos [6]. Este tipo de redes cuentan con una velocidad de aprendizaje más rápida, lo que representa una gran ventaja.

En una red neuronal RBF, las neuronas en la capa oculta utilizan funciones radiales para medir la distancia existente entre los datos de entrada y ciertos puntos de referencia o centros. Estos centros representan puntos significativos y se utilizan para ponderar la influencia que tiene cada neurona de la capa oculta en la predicción o clasificación final. En la figura 1 se presenta un esquema de las redes neuronales RBF, en general, siguen la siguiente arquitectura [6, 7]:

1. Capa de entrada: En esta primera capa, los datos de entrada son proporcionados a la red.
2. Capa oculta: Se encuentra interconectada entre todos sus nodos con la capa de entrada y es activada a través de la función radial (gaussiana).

3. Capa de salida: Una vez que las neuronas de la capa oculta se activan, sus salidas se combinan ponderadamente para formar la salida de la red. Esta capa es activada a través de una función lineal continua.

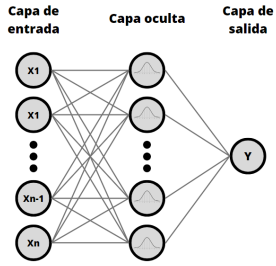


Figura 1. Esquema de una red neuronal de función base radial.

Existen múltiples sistemas dinámicos cuyos modelos matemáticos se caracterizan por ser de primer orden. En algunos casos las herramientas clásicas no logran un sistema de control que sea estable, posea un buen desempeño y rechace adecuadamente perturbaciones [8]. Una solución prometedora son los controladores por modos deslizantes (SMC, por sus siglas en ingles), es uno de los métodos de control robusto y no lineal, tiene varias ventajas tales como la robustez contra perturbaciones externas, sin embargo, su principal desventaja es que agrega un término discontinuo en el controlador y la discontinuidad da como resultado una frecuencia de conmutación infinita, lo que se ve reflejado en el consumo energético de este tipo de controladores [9-11]. Es por ello que los SMC son inviables si lo que se busca es reducir el costo computacional y energético.

Las redes neuronales RBF fueron propuestas a finales de los años 1980, pero gracias a la creciente popularidad de la inteligencia artificial han tomado mayor relevancia en la actualidad. Existen diferentes publicaciones que incorporan las RBF, sin embargo, la mayoría de ellas solo son empleadas de manera complementaria. En [5] y [12] se realiza una implementación hibrida con un control por modo deslizante adaptativo mediante redes neuronales RBF, lo cual mejora el tiempo de convergencia. Por otro lado, en [13] y [14] se propone una variante de PID adaptativo, el cual hace uso de redes neuronales RBF para realizar la sintonía de las ganancias. Estas propuestas [5, 12-14] mantienen una implementación complementaria, por lo que se requiere un módulo de control y un módulo que contiene la red neuronal RBF. En este artículo se propone una red neuronal RBF, la cual sustituye por completo el módulo de control, simplificando el sistema y aprovechando los beneficios que poseen las RBF.

3 Diseño del control neuronal adaptativo

3.1 Planteamiento del problema

Considerando un sistema dinámico de primer orden, representado en la expresión (1)

$$\dot{x}(t) = f\big(x(t)\big) + u(t) + \varsigma(t), \tag{1}$$

donde $f(\cdot) \in \mathbb{R}$ representa la dinámica no lineal del sistema, que en este trabajo se denota como una incertidumbre dependiente del estado, $\varsigma(t) \in \mathbb{R}$ es una perturbación exógena variante en el tiempo, sabiendo que $\dot{\varsigma}(t) \neq 0$, la cual es desconocida, pero se puede delimitar a $\|\dot{\varsigma}_i(t)\| \leq \varsigma_{max}$, y finalmente, $u(t) \in \mathbb{R}$ es la entrada de control.

3.2 Aproximación basada en redes neuronales

Dado que las dinámicas no lineales $f(x)$ y $g(x)$ son desconocidas, por lo tanto, una combinación lineal finita de funciones de activación ponderadas para aproximar dichos términos no lineales dentro de un conjunto prescrito $\boldsymbol{\Omega}$.

Teorema 3.2.1 (Aproximación universal) Sea $f(x(t))$ una función suave en el conjunto compacto Ω. Por lo tanto, existe una arquitectura de red neuronal de tres capas basada en una única capa intermedia de η neuronas que presenta una función de activación adecuada $\phi\big(x(t)\big) \in^{\rho}$ y un vector de pesos ideales $\omega \in \mathbb{R}^{\rho}$, tal que puede ser descrita por la ecuación (2)

$$f(x) = \boldsymbol{\omega}^T \boldsymbol{\varphi}\big(x(t)\big) + \varepsilon, \tag{2}$$

donde ε representa el error de aproximación de la RNA. En cuanto a la función de activación, se puede seleccionar entre una tangente hiperbólica, sigmoide o gaussiana (RBF).

Suposición 3.2.2 El peso ideal $\omega(\cdot)$, la función de activación $\varphi(x)$ y el error de aproximación ε tienen su límite superior respectivamente en $\|\omega(\cdot)\| \leq \omega_{max}$, $\|\varphi(\cdot)\| \leq \varphi_{max}$ y $\|\varepsilon\| \leq \varepsilon_{max}$, con $\omega_{max}, \varphi_{max}, \varepsilon_{max} > 0 \in \mathbb{R}$.

Observación 3.2.3 El error de aproximación desconocido ε_i puede ser reducido incrementando el número de neuronas. El teorema 3.2.1 permite aproximar uniformemente $f(\xi)$ usando la expresión (3):

$$\hat{f}\big(x(t)\big) = \hat{\boldsymbol{\omega}}^T \boldsymbol{\varphi}(x(t)), \tag{3}$$

Donde $\boldsymbol{\omega} = \big[\omega_1, \dots, \omega_\rho\big]^T$ y $\boldsymbol{\varphi} = \big[\varphi_1, \dots, \varphi_\rho\big]^T$, donde ρ representa el número de neuronas. En este trabajo, consideramos funciones de base radial (RBF) como función de activación. En la ecuación (4) se presenta la función de activación utilizada.

$$\varphi = \exp\left(-\frac{(x-\mu)^2}{\sigma^2}\right), \tag{4}$$

donde μ y σ representan la media y la varianza de las entradas (x).

3.3 Diseño del controlador

La política de control propuesta pretende alcanzar el objetivo de regulación mientras compensa la dinámica desconocida dependiente del estado, $f(x)$ y las perturbaciones externas $\varsigma(t)$. En otras palabras, el esquema de control de primer orden pretende minimizar el error de posición, como se representa en la expresión (5)

$$\lim_{t\to\infty} \|x(t) - x_d(t)\| \to 0, \tag{5}$$

En ese sentido, reescribimos el modelo en una dinámica de error de primer orden, obteniendo la ecuación (6)

$$\dot{e}(t) = f\big(x(t)\big) + u(t) + \varsigma(t) - \dot{x}_d(t), \tag{6}$$

donde $e = x(t) - x_d(t)$. En cuanto al diseño del control consideramos el termino global representado por la expresión (7)

$$\bar{u}(t) = f\big(x(t)\big) + u(t) + \varsigma(t), \tag{7}$$

lo que permite reescribir la dinámica de error anterior, obteniendo la ecuación (8)

$$\dot{e}(t) = \bar{u}(t) - \dot{x}_d(t). \tag{8}$$

El controlador propuesto para estabilizar (6) se encuentra dado por la expresión (9):

$$\bar{u}(t) = -\widehat{\boldsymbol{\omega}}^T \boldsymbol{\varphi}\big(e(t)\big) + \varepsilon + \dot{x}_d(t) \tag{9}$$

y cuya regla de adaptación/aprendizaje se representa por la ecuación (10):

$$\dot{\hat{\omega}}(t) = -\alpha e(t)\varphi\big(e(t)\big), \tag{10}$$

donde $\alpha \in \mathbb{R}$ es la tasa de aprendizaje.

Observación 3.3.1 El peso estimado se calcula a partir de la minimización del error de estimación del peso, expresado en la ecuación (11)

$$\tilde{\omega} = \omega - \hat{\omega}. \tag{11}$$

Su derivada temporal correspondiente se presenta en la ecuación (12)

$$\dot{\tilde{\omega}} = -\dot{\hat{\omega}} \tag{12}$$

3.4 Análisis de estabilidad

Sea una función candidata de Lyapunov dada por la ecuación (13)

$$V = \frac{1}{2}e(t)^2 + \frac{1}{2\alpha}\tilde{\boldsymbol{\omega}}^T(t)\tilde{\boldsymbol{\omega}}(\boldsymbol{t}). \tag{13}$$

Diferenciando (13) con respecto al tiempo y considerando (12), obtenemos la expresión (14)

$$\dot{V} = e(t)\dot{e}(t) - \frac{1}{\alpha}\tilde{\boldsymbol{\omega}}^T(t)\dot{\hat{\boldsymbol{\omega}}}(t). \tag{14}$$

Considerando (8) y (9), podemos rescribir (14) de la siguiente forma, presentada en la ecuación (15)

$$\dot{V} = e(t)(-\boldsymbol{\omega}^T\boldsymbol{\varphi}(e(t)) + \varepsilon) - \frac{1}{\alpha}\tilde{\boldsymbol{\omega}}^T(t)\dot{\hat{\boldsymbol{\omega}}}(t). \tag{15}$$

Usando (11), obtenemos la expresión (16)

$$\dot{V} = -e(t)\widehat{\boldsymbol{\omega}}^T\boldsymbol{\varphi}\big(e(t)\big) - e(t)\tilde{\boldsymbol{\omega}}^T\boldsymbol{\varphi}\big(e(t)\big) + e(t)\varepsilon - \frac{1}{\alpha}\tilde{\boldsymbol{\omega}}^T(t)\dot{\hat{\boldsymbol{\omega}}}(t), \tag{16}$$

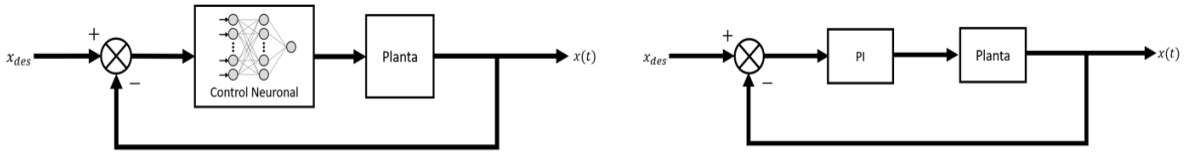
donde hemos sustituido $\omega = \tilde{\omega} + \hat{\omega}$. Ahora utilizando (12) se obtiene la ecuación (17)

$$\dot{V} = -e(t)\|\widehat{\boldsymbol{\omega}}^T\|\big\|\boldsymbol{\varphi}\big(e(t)\big)\big\| + e(t)\varepsilon, \tag{17}$$

considerando que podemos despreciar a ε ya que se reduce a medida que el número de neuronas aumenta y $\|\widehat{\boldsymbol{\omega}}^T\|, \|\boldsymbol{\varphi}(e(t))\| > 0$, permiten concluir que el vector de pesos de error $\widetilde{\boldsymbol{\omega}}$ y el error $e(t)$ están y permanecen acotados dentro de una vecindad de origen para $t > 0$.

4 Simulación del control neuronal adaptativo

Partiendo de un sistema dinámico de primer orden, la red neuronal de base radial es diseñada para sustituir a un controlador clásico, por lo que utiliza como variable principal el error de posición, es decir la diferencia entre la posición deseada y la posición actual ($e = x_d - x(t)$). La red neuronal que se utiliza se encuentra conformada por un total de 600 neuronas y los pesos iniciales son asignados de manera aleatoria. Esta red realiza un aprendizaje en línea, por lo que no necesita de una etapa de entrenamiento previa o un conjunto de datos, esto permite que aún con el cambio de las perturbaciones el sistema siga adaptándose, minimizando el error a valores muy cercanos a cero. El control que se utiliza con fines comparativos es un PI clásico, debido al buen desempeño que lo caracteriza y por ser ampliamente utilizado. En la figura 2 se presenta un diagrama a bloques del lazo cerrado de control tanto del controlador PI clásico como del control neuronal adaptativo.



a) Control Neuronal Adaptativo b) Control PI Clásico

Figura 2. Diagrama a bloques del control neuronal adaptativo y el control PI clásico.

Para la simulación, se utilizó el software MATLAB – Simulink para obtener la respuesta de los controladores en presencia de diferentes tipos de perturbaciones. Los casos analizados son; perturbación contante ($\dot{x}(t) = u(t) + \delta$), perturbación variante en el tiempo ($\dot{x}(t) = u(t) + \varsigma(t)$) y perturbación dependiente del estado ($\dot{x}(t) = u(t) + \varsigma(t) + f(x(t))$). A continuación, en la figura 3 se presenta el diagrama a bloques implementado en Simulink para la simulación de los 3 casos.

5 Resultados

A continuación, se describen y presentan los resultados correspondientes a cada una de las incertidumbres anteriormente mencionadas. Se considera una posición deseada $x_d = 10$ y una condición inicial $x_0 = -5$.

5.1 Perturbación constante

En la figura 4 se puede visualizar la respuesta para el sistema de la forma $\dot{x}(t) = u(t) + \delta$, con una perturbación constante $\delta = 5$ a partir del tiempo $t = 10$. La primera gráfica representa la posición a través del tiempo y se puede observar que el control neuronal adaptativo reacciona mucho más rápido que el PI clásico, disminuyendo el error a prácticamente 0 a partir del segundo 1.7; posterior a la perturbación, vuelve a alcanzar la posición deseada alrededor de los 11 segundos, mientras que el PI nunca converge completamente. La segunda gráfica corresponde al error de posición y se puede observar con mayor precisión que el control neuronal adaptativo logra un error exactamente igual a 0, mientras que el PI clásico nunca alcanza la posición deseada.

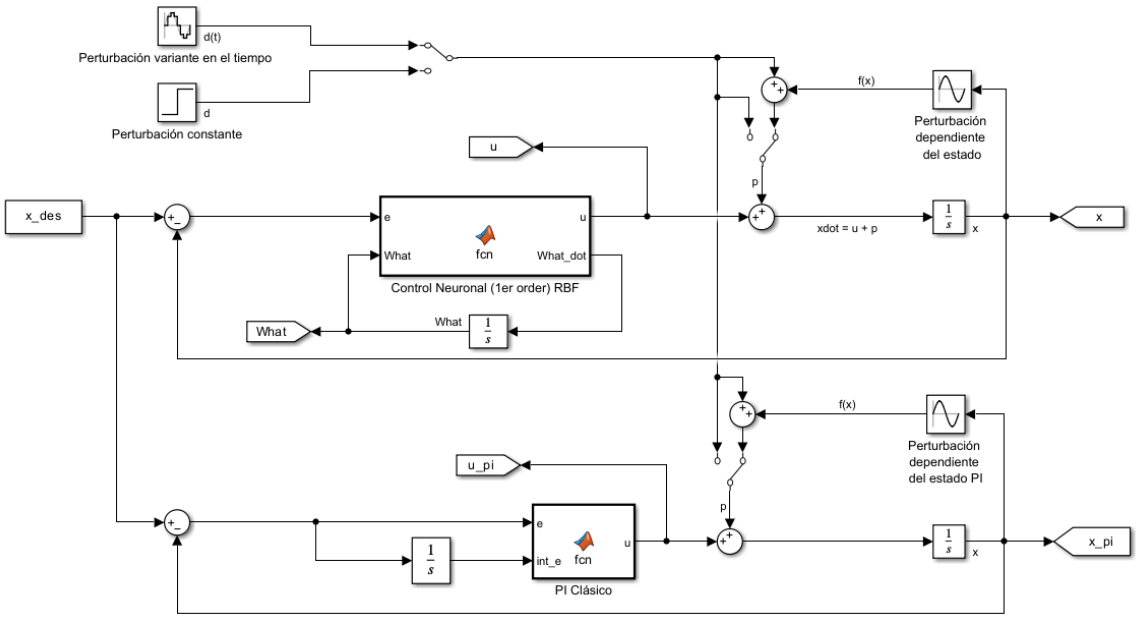


Figura 3. Diagrama a bloques en MATLAB – Simulink.

5.2 Perturbación variante en el tiempo

En la figura 5 se puede visualizar la respuesta para el sistema de la forma $\dot{x}(t) = u(t) + \varsigma(t)$, con una perturbación variante en el tiempo $\varsigma(t) = 10\sin(2t)$. La primera gráfica representa la posición a través del tiempo, se puede observar que el control neuronal adaptativo disminuye el error rápidamente, con oscilaciones mínimas a partir del segundo 1.1, por otro lado, el PI clásico inicia la simulación con oscilaciones menores, sin embargo, al pasar el tiempo comienza a incrementar la amplitud de sus oscilaciones, esto representa un problema debido a que es una perturbación que depende del tiempo. La segunda gráfica corresponde al error de posición y se puede observar una menor amplitud de las oscilaciones del control neuronal adaptativo, además, tomando como referencia el segundo 18, el control neuronal presenta un error del 3.2% mientras que el PI clásico tiene un error del 10% en el mismo instante de tiempo.

5.3 Perturbación dependiente del estado

En la figura 6 y 7 se puede visualizar la respuesta para el sistema de la forma $\dot{x}(t) = u(t) + \varsigma(t) + f(x(t))$. En la figura 6 con una perturbación variante en el tiempo $\varsigma(t) = 10\sin(2t)$ y otra dependiente del estado $f(x(t)) = 4\sin(3x(t))$, la primera gráfica representa la posición a través del tiempo, se puede observar que el control neuronal adaptativo disminuye el error rápidamente y así permanece, mientras que el PI clásico incrementa el error conforme transcurre el tiempo. La segunda gráfica corresponde al error de posición, en donde se puede observar una menor amplitud de las oscilaciones por parte del control neuronal, considerando un tiempo igual a 3.2 segundos, el control neuronal presenta un del 3.5%, el cual sigue disminuyendo, mientras que el PI clásico tiene un error del 5.2%, el cual se mantiene incrementando. Finalmente, en la figura 7, se incrementa la perturbación dependiente del estado a $f(x(t)) = 6\sin(3x(t))$ y se logra visualizar una mayor diferencia en la gráfica del error de posición, en donde, en PI presenta una oscilación máxima del 13.9% en el segundo 19.6, mientras que el control neuronal adaptativo presenta una oscilación máxima del 5% en el segundo 19.1.

Es bien sabido que el controlador PI clásico es ampliamente utilizado debido a su buen desempeño, ya que elimina el error en el estado estacionario y generalmente es un referente para comparar cualquier otra estrategia de control; sin embargo, se puede observar que el control neuronal adaptativo logra alcanzar errores aún menores. En general, se observa una respuesta bastante rápida por parte del control neuronal para todos los casos analizados. Al obtener una salida continua y alcanzar la posición deseada en un menor tiempo, se pueden reducir los tiempos de simulación y costos computacionales. La simulación se realizó sustituyendo directamente el bloque de control por la red neuronal RBF, lo que simplifica su implementación, siendo este otro de los principales beneficios de la propuesta frente a los sistemas actuales.

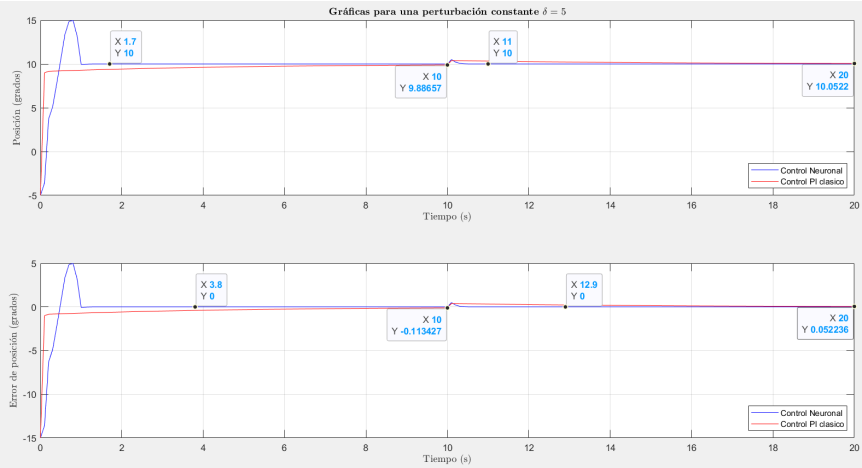


Figura 4. Resultados obtenidos para una perturbación constante $\delta = 5$ a partir del tiempo $t = 10$.

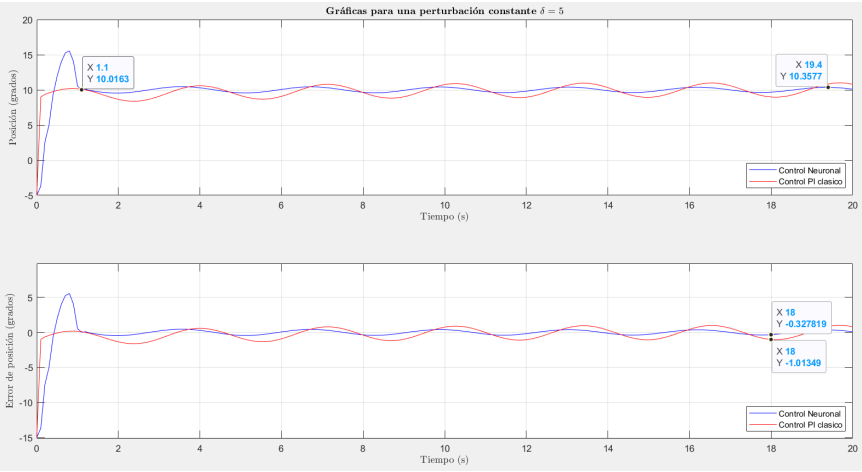


Figura 5. Resultados obtenidos para una perturbación variante en el tiempo $\varsigma(t) = 10\,\mathrm{sin}(2t)$.

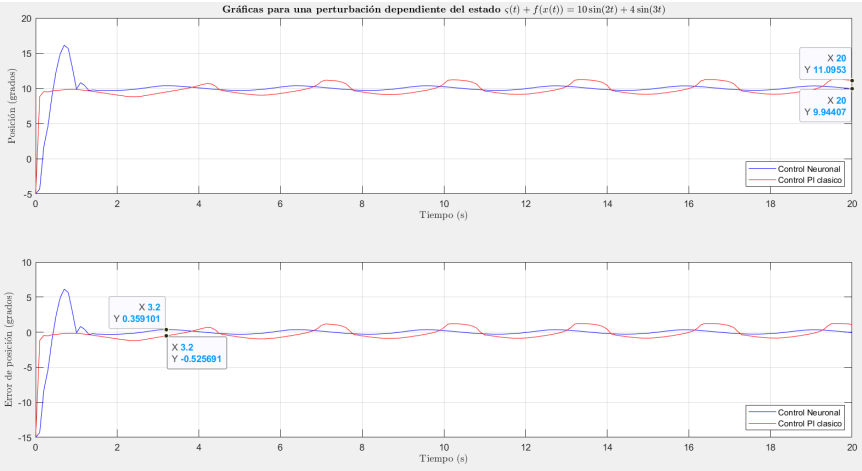


Figura 6. Resultados obtenidos para una perturbación dependiente del estado $\varsigma(t) + f(x(t)) = 10\sin(2t) + 4\sin(3x)$.

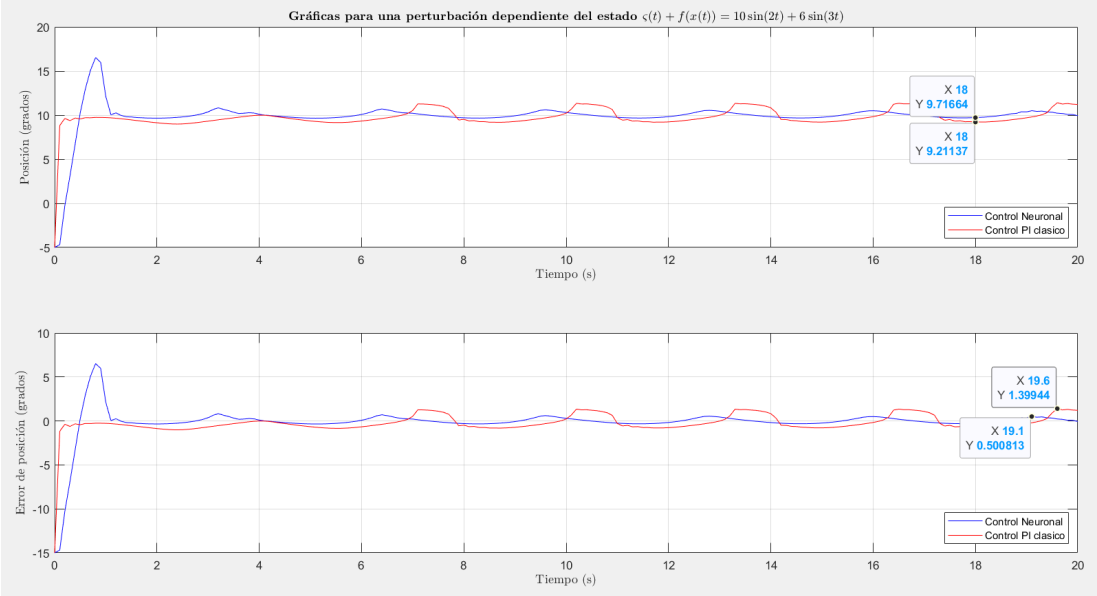


Figura 7. Resultados obtenidos para una perturbación dependiente del estado $\varsigma(t) + f(x(t)) = 10\sin(2t) + 6\sin(3x)$.

6 Conclusiones y trabajo a futuro

El diseño del control neuronal adaptativo se logró gracias al estudio de las redes neuronales de función base radial. Utilizando el software de simulación Matlab – Simulink, se obtuvo el comportamiento de un sistema dinámico de primer orden al aplicar un control neuronal adaptativo y se comparó con un control PI clásico en presencia de diferentes tipos de incertidumbres, abarcando los casos; perturbación contante, perturbación variante en el tiempo y perturbación dependiente del estado. El control neuronal adaptativo resultó ser más eficiente, pues le lleva un menor tiempo alcanzar la posición deseada x_d, además el error es significativamente menor en el caso de la perturbación constante. Por otro lado, en el caso de perturbaciones dependiente del tiempo y dependiente del estado, se pueden observar oscilaciones menores, las cuales disminuyen

su amplitud conforme transcurre el tiempo para el control neuronal. Algunas de las deficiencias del control neuronal adaptativo es que en todos los casos se presenta un sobre impulso bastante grande, lo que podría hacerlo poco viable para determinadas aplicaciones, además, al realizar un entrenamiento en línea, los resultados no serán siempre los mismos, las variaciones son mínimas, sin embargo, se pueden obtener mejores o peores resultados. De manera general, este control neuronal adaptativo, es capaz de alcanzar el valor deseado de una manera más rápida, lo que permite un menor tiempo de simulación y por ende un menor costo computacional, algo que sin duda impacta en el consumo energético, lo cual representa una gran ventaja en la implementación de redes neuronales en general. El control propuesto realiza una adaptación (aprendizaje) en línea, por lo que no necesita de una etapa de entrenamiento previa, esto permite que aún con el cambio de las perturbaciones el sistema siga adaptándose y consiguiendo errores muy cercanos a cero. Este trabajo tiene la capacidad de ser escalable, es decir, posteriormente se pretende realizar la simulación para sistemas de segundo orden y así poder visualizar el comportamiento de esta nueva familia de controladores neuronales adaptativos ante sistemas más complejos.

Agradecimientos

Los autores de este artículo agradecen al Consejo Nacional de Ciencia y Tecnología (CONACYT) y a la Benemérita Universidad Autónoma de Puebla (BUAP) por el apoyo otorgado en la realización de este trabajo.

Referencias

[1] P. Galeone. “Hands-on neural networks with TensorFlow 2.0: understand TensorFlow, from static graph to eager execution, and design neural networks”. Packt Publishing Ltd, 2019.

[2] L. Shuai, et al. "Neural Networks for Robot Arm Cooperation with a Full Distributed Control Topology". Neural Networks for Cooperative Control of Multiple Robot Arms (2018): 49-74.

[3] M. Revanesh, et al. "Artificial neural networks-based improved Levenberg–Marquardt neural network for energy efficiency and anomaly detection in WSN". Wireless Networks (2023): 1-16.

[4] L. Shengnan, et al. "Energy efficiency and coding of neural network". Frontiers in Neuroscience 16 (2023).

[5] X. ZHANG, et al. "Research on fixed time sliding mode control strategy based on RBF neural network". Journal of Measurement Science & Instrumentation 14.2 (2023).

[6] H. Wang and F. Meng. "Control Method of Nanomaterial Numerical Control Electronic Processing Based on RBF Neural Network". Advances in Materials Science and Engineering 2022 (2022).

[7] A. Doug. “Neural networks: history and applications”. Nova Science Publishers, Incorporated, 2020.

[8] O. Regalón, et al. "Aplicación de algoritmos de control clásico, adaptable y robusto a sistemas dinámicos de parámetros variables". Ingeniería Energética 33.3 (2012): 184-195.

[9] A. Panhale, et al. "Robust motion control using novel first order sliding modes." 2020 20th International Conference on Control, Automation and Systems (ICCAS). IEEE, 2020.

[10] V. Utkin, et al. “Sliding mode control in electro-mechanical systems”. CRC press, 2017.

[11] C. Pérez-Pirela, et al. "Control por modos deslizantes de un sistema de intercambio de calor: validación experimental." Enfoque UTE 9.4 (2018): 110-119.

[12] H. Tian, et al. "Research on Adaptive Sliding Mode Robust Control Algorithm of Manipulator Based on RBF Neural Network." 2020 Chinese Automation Congress (CAC). IEEE, 2020.

[13] Y. Jing, et al. "Inverted pendulum RBF neural network PID controller design". 2014 International Symposium on Computer, Consumer and Control. IEEE, 2014.

[14] R. Wang, et al. "Fuzzy neural network PID control based on RBF neural network for variable configuration spacecraft". 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). IEEE, 2018.

Comparativa de herramientas para la identificación de vulnerabilidades en AlmaLinux …

Comparison of tools for the identification of vulnerabilities in AlmaLinux …

Yeiny Romero Hernández, Judith Pérez Marcial, María del Carmen Santiago Díaz, Gustavo Trinidad Rubín Linares, Ana Claudia Zenteno Vázquez, Rosa Isabel Pérez Ortega
Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla,
Av. San Claudio y 14 sur, Col. San Manuel C.P. 72000, Puebla, Puebla, México.
{yeiny.romero, judith.perez, marycarmen.santiago, gustavo.rubin,
ana.zenteno}@correo.buap.mx, rosa.perezor@alumno.buap.mx

Abstract

Cybersecurity is something important in the world of the Internet, given that the information circulating in the cloud is vulnerable without correct protection and can fall into extortion and theft of personal data among the minor damages, which is why it is important to know that there is Ethical hacking that helps test tools in order to detect vulnerabilities and protect equipment, which is why the objective of this work is to test vulnerability detection tools to verify if we can complement or determine the best one for protection. of our data on the network.

Resumen

La ciberseguridad es algo importante en el mundo del internet, dado que la información circulante en la nube es vulnerable sin una protección correcta y se puede caer en extorciones y robo de datos personales entre los menores daños, por lo cual, es importante conocer que exist eel hacking ético que ayuda a probar herramientas con el fin de detector las vulnerabilidades y proteger los equipos, es por ello, que el objetivo de este trabajo es probar herramientas de detección de vulnerabilidades para verificar si las podemos complementar o determiner la major para la portección de nuestros datos en la red.

Keywords and phrases: Ciberseguridad, Hacking Ético, Herramientas, Vulnerabilidades..

1 Introducción

La ciberseguridad tiene distintas definiciones, pero de manera general que hace referencia al cómo protegemos los datos de las amenazas que puedan poner en riesgo la información en el ciberespacio.[1][2]

1.1 Ramas de la ciberseguridad

Existen diferentes ramas que abarcan los aspectos de la ciberseguridad como son: el hacking ético, el cómputo forense, las auditorías de seguridad digital, la seguridad en las redes informáticas, el peritaje judicial, el big data en entornos seguros, la investigación y la persecución del cibercrimen entre otras.[3][4]

1.2 Hacking ético

Una de las ramas más desarrolladas por la importancia de la protección de datos, expertos en seguridad de los sistemas de información están dedicados a realizar pruebas de penetración en busca de alguna vulnerabilidad o falla en los sistemas instalados y en funcionamiento.[5]

Se le conoce como hacker ético a la persona que utiliza sus conocimientos en programación, redes de computadoras, entro otros para realizar pruebas de penetración de forma controlada con el respectivo permiso sobre la infraestructura y los sistemas de información, esto sin hacer ningún daño.[5][6]

Existen diferentes fases de una prueba de penetración de entre los cuales tenemos de manera generalizada de la Fase de informe: En esta fase podemos encontrarnos con la elaboración del informe indicando las vulnerabilidades que fueron encontradas además de que se documenta el cómo fueron explotadas.[7]

1.3 Herramientas de vulnerabilidades

Definamos primero lo que es una vulnerabilidad la cual hace referencia a los errores, fallos o huecos de seguridad que están presentes en un software, plataforma o programa informático, los cuales pueden o no pasar desapercibidos por los programadores o administradores del sistema pero que los atacantes o cibercriminales pueden identificar y a través de diferentes ataques pueden ocasionar actividades que crean riesgos en los sistemas además de también generar impactos negativos en estos.[8]

De entre las herramientas más utilizadas para la detección de vulnerabilidades tenemos:

- QualysGuard Web Application Scanning WAS: Esta herramienta, la cual está en la nube, nos permite realizar pruebas con selenium para aplicaciones web, además de poder realizar pruebas de penetración. Con esta herramienta podemos encontrar vulnerabilidades del top 10 de OWASP.
- WEBAPP 360: Enterprise Class web application scanning: Con esta herramienta podemos evaluar de una forma completa la infraestructura de aplicaciones web incluyendo sistemas operativos subyacentes además de aplicaciones subyacentes en entorno de producción.
- Parasoft C/C++ Test: La podemos utilizar en pruebas para aplicaciones basadas en C y C++, esta herramienta ayuda a los desarrolladores a la prevención y eliminación de defectos, pudiendo eliminar los problemas de seguridad.
- Nessus Vulnerability Scanner: Con esta herramienta podemos escanear las vulnerabilidades en servidores web, servicios web, además de las vulnerabilidades de OWASP.[9]
- Nikto: Es una herramienta la cual nos permite identificar archivos que sean potencialmente peligrosos. Esto lo logra gracias a que utiliza un registro de vulnerabilidades el cual contiene

información para que durante el escaneo pueda reconocer las amenazas a las que puede estar expuesta la información.[10][11]

Algo a tomar en cuenta es que Nikto no fue diseñado para ser una herramienta no intrusiva, si no que intenta evaluar el servidor web en el menor tiempo, siendo bastante obvio en los archivos de registro o log files.

Para su instalación se puede descargar directamente utilizando la herramienta APT o puede ser instalada desde su sitio web. [12][13]

En lo que se refiere al soporte SSL se debe tener instalada la librería OpenSSL además del módulo Net: SSLeay de perl. Estas librerías pueden ser instaladas fácilmente utilizando apt-get.[10]

- Vega: Es un escáner de seguridad web gratuito y de código abierto. Esta herramienta nos permite realizar pruebas de seguridad en aplicaciones web. Algunas de las pruebas que podemos realizar tenemos el poder encontrar vulnerabilidades tales como inyecciones SQL, Cross-Site Scripting (XSS), información confidencial que pudo ser revelada de manera inadvertida además de otras vulnerabilidades más.[14] [15] Esta herramienta está escrita en Java y está disponible para Linux, OS X y Windows.[16]

- Acunetix: Es una herramienta la cual nos sirve para realizar un escaneo sobre las vulnerabilidades web. Esta herramienta es utilizada por diversas empresas y es muy aclamado ya que incluye la inyección SQL más avanzada además de tener la tecnología de escaneo de caja negra XSS.[17] Esta herramienta clasifica los diferentes riesgos de seguridad en una escala de entre alto, medio y bajo, además de que nos proporciona una interfaz gráfica y reportas con medidas de solución.[18]

Esta herramienta puede realizar un escaneo a cualquier sitio web mediante el protocolo HTTP/HTTPS.[19]

- Ratproxy: Es una herramienta la cual nos permite la buscar vulnerabilidades web. Esta herramienta es capaz de identificar los datos que se transportan por el SSL. Lo que hace es examinar las respuestas JSON sospechosas, pudiendo saber qué tipo de datos existen en el caché.[20] Esta herramienta es capaz de distinguir entre las hojas de estilos CSS y los códigos JavaScript.[21]

- Wfuzz: Esta herramienta está diseñada para las aplicaciones web de fuerza bruta ya que nos ayuda a encontrar los recursos que no estén vinculados, parámetros GET y POST por medio de la fuerza bruta para poder verificar diferentes tipos de inyecciones.[22][23]. Wfuzz está basado en el fuzzing, la cual es una técnica que interroga al servidor por la existencia de archivos o directorios que contengan las palabras que están incluidas en el diccionario previamente seleccionado para la prueba.[24]

- John the Ripper: Esta es una herramienta útil en la recuperación de contraseñas, con la cual podremos realizar auditorías de seguridad de contraseñas. Está disponible en varios sistemas operativos como lo son Linux, Windows y Mac OS X, ofreciéndonos versiones tanto en línea de comandos como con GUI.[25][26][27]

Una vez que ya conocimos la mayor parte de los conceptos y sabemos que debemos protegernos de las vulnerabilidades, debemos explorar algunas de las herramientas de detección de vulnerabilidades y probarlas con la intención de identificar que tanto nos pueden proteger de los ataques que puedan existir y finalmente realizar una comparativa entre las herramientas seleccionadas para saber cuál herramienta es la mejor.

2 Metodología

Conociendo el conjunto de herramientas disponibles hemos seleccionado al menos 3 de ellas para probarlas y determinar de qué manera pueden ayudarnos a proteger la información en la nube. Dichas herramientas fueron Nessus, Nikto y John the Ripper.

Nessus

Comencemos con la herramienta llamada Nessus. Esta herramienta puede ser ejecuta desde la consola o con una interfaz gráfica. De entre las ventajas que nos proporciona Nessus tenemos:

- Se obtiene una respuesta inmediata al momento que va detectando puertos abiertos.
- Tiene un control contra la detección de hackers.
- Tiene un clasificador de riesgos (bajo, medio y alto) para cada amenaza detectada.
- Tiene una detección automática de protocolos.
- Tiene una detección de alertas basadas en scripts y en reglas.

En cuanto a las desventajas tenemos que es una herramienta la cual para las organizaciones podría costar algo de dinero ya que tiene una versión de paga. También tiene una versión gratuita, pero está limitada a ciertas acciones o plugins que la versión de paga contiene y para una organización son muy útiles.[28][29]

Nessus está dividida en dos componentes, la parte del servidor nessusd, encargada de realizar las pruebas o tests, y el cliente el cual es quien se encarga de la interfaz con el usuario pudiendo estar en otra máquina diferente.[30]

Primeramente, debemos realizar actualizaciones y descargar paquetes que nos apoyen a la descarga de la herramienta Nessus, una vez que descarguemos e instalemos el paquete procederemos a inicializar el servicio de nessusd. Luego de esto de le daremos permisos adecuados al firewall para que pueda funcionar sin problemas (ver Figura 1).

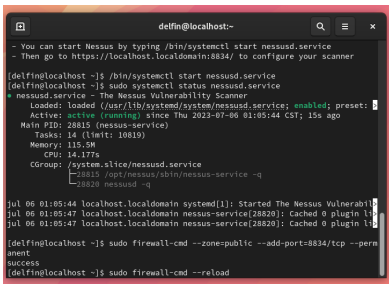


Figura 1. Comando para configurar los permisos de firewall para la herramienta Nessus

Lo siguiente será ingresar en nuestro navegador a la dirección y puerto correspondiente con el protocolo https. De manera general, es muy intuitivo, por lo que fácilmente crearemos una cuenta con la que posteriormente accederemos a la herramienta.

Nikto

La siguiente herramienta que se probará será Nikto la cual es una herramienta opens source gratuito. De entre las ventajas que tiene esta herramienta en servidores web es que puede realizar test exhaustivos los cuales contienen más de 3,200 ficheros.

Nikto puede emitir el reporte de los hallazgos en su escaneo en diferentes formatos. Además, nos proporcionará sugerencias de cómo arreglarlos. Una desventaja que puede tener es que al utilizarse mediante consola puede resultar un poco difícil de utilizar especialmente para detecciones más avanzadas.[10]

En cuanto a la instalación de Nikto, tenemos que lo primero que haremos será actualizar el sistema operativo, luego de eso instalaremos epel-release para poder instalar herramientas de terceros ya que la instalaremos desde su repositorio oficial en GitHub.

Luego procederemos a instalar Perl (ver Figura 2), ya que será necesario para poder realizar los escaneos con la herramienta Nikto.

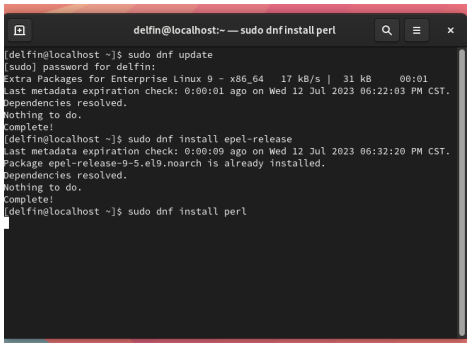


Figura 2. Instalación de Perl, epel-release y actualización del sistema operativo para instalación de Nikto

Luego vamos a clonar el repositorio de la herramienta Nikto En este caso Alma Linux no cuenta con git por lo que automáticamente lo detectará y nos dirá si queremos instalarla a lo que diremos que sí. Una vez clonado el repositorio procederemos a acceder al script principal, ahora estaremos listos para poder ejecutar el escaneo por medio de comandos.

John the Ripper

Por último, tenemos a John the Ripper, una herramienta muy útil en la recuperación y evaluación de contraseñas. Esta herramienta nos ofrece una versión tanto en línea de comandos como con interfaz gráfica.

El tener una contraseña en un sistema puede ser de gran impacto ya que, si por ejemplo alguien lograse obtener nuestra contraseña del administrador, con ella podría comprometer el sistema gravemente, pudiendo robar mucha información confidencial, por lo tanto esta herramienta nos ayudarán a recuperar contraseñas y con esto podremos auditar las contraseñas que se tienen en nuestro sistema, para que en caso de que se detecte alguna contraseña débil, sea notificada a la persona respectiva y con esto reducir algún posible acceso al sistema de alguien no autorizado.

De entre las opciones que tenemos está el elegir el modo de ataque, aplicar una máscara, aplicar ciertas reglas, poder crear sesiones, las cuales son útiles para cuando se necesita parar el ataque y posteriormente reanudarlo. También podemos revisar el rendimiento de la herramienta para cada algoritmo hash utilizado, además de la creación de procesos.[31]

3 Resultados

Comencemos con Nessus desde el navegador en la dirección https://localhost:8834, una vez ahí nos pedirá nuestra cuenta y podremos acceder

Durante el escaneo podremos ver las vulnerabilidades encontradas, lo cual ayuda mucho a no esperar al final para saber cuáles vulnerabilidades son las que se encontraron, luego tenemos otro apartado donde vemos las vulnerabilidades encontradas Finalmente tenemos un historial de los escaneos realizados. Además del apartado de las soluciones, podemos ver de manera individual una descripción de la vulnerabilidad (ver Figura 3).

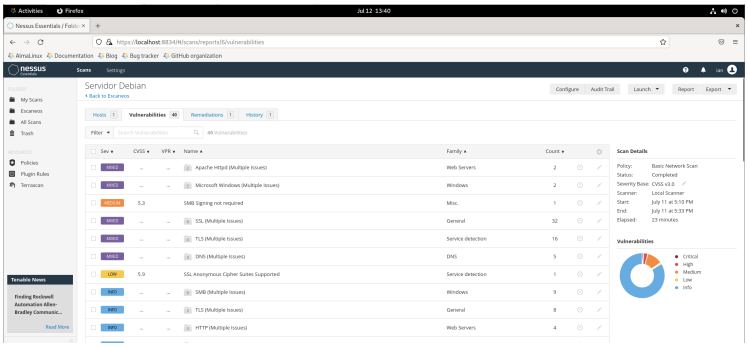


Figura 3. Apartado de vulnerabilidades encontradas utilizando la herramienta Nessus.

Nikto

Cambiaremos la dirección de ejemplo por la dirección objetivo a la que buscaremos las vulnerabilidades. Por defecto utiliza el puerto 80 buscar las vulnerabilidades (ver Figura 4).

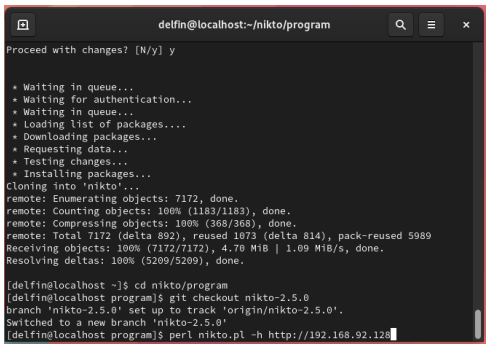


Figura 4. Escaneo con la herramienta Nikto

Podemos guardar los resultados en distintos formatos o simplemente ver los resultados por medio de la consola. Con ello facilitamos la lectura de sus resultados.

John the Ripper

Procederemos a obtener las contraseñas cifradas las cuales están en la ruta /etc/shadow. Se ingresan en un archivo de prueba y se ejecuta la herramienta. Otras pruebas realizadas fueron utilizando solo minúsculas, solo mayúsculas y solo números. Para la prueba de solo números se utilizaron dos números distintos, además de que se utilizó la opción john --incremental:Digits contras.txt para que solo intentara números y no otras combinaciones. (ver Figura 5)

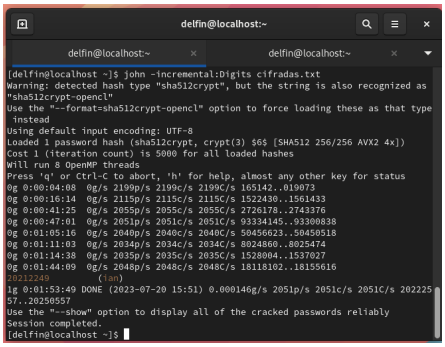


Figura 5 Contraseña encontrada utilizando solo números

5 Conclusiones y trabajo a futuro

Nessus fue muy efectiva encontrando vulnerabilidades ya que encontró un buen número de vulnerabilidades al escanear el servidor, sin embargo, es una herramienta de paga que si deseamos proteger correctamente un servidor escolar o empresarial debemos considerar en la inversión. La herramienta Nikto también logró encontrar vulnerabilidades web, aunque sin duda Nessus sería más optimo debido a que muestra más información de la vulnerabilidad además de la solución a la misma.

Debemos tener en cuenta que Nessus escanea el servidor completo y Nikto solo escanea los servidores web.

Finalmente, la herramienta John the Ripper, demostró ser muy efectiva en encontrar contraseñas, se probaron diversas contraseñas en apariencia de diversas combinaciones y todas fueron encontradas, pero utilizando las recomendaciones de contraseñas seguras, se lleva a cabo el uso de mayúsculas, minúsculas, números y caracteres especiales en una sola combinación y por tanto, se logró que no se vulnerara esta contraseña.

Podemos concluir que la mezcla de herramientas protege de mejor manera un servidor, es decir, utilizando Nessus con John the Ripper se lográ una mejor detección de vulnerabilidades, lo que ayuda a proteger de una manera eficiente cualquier servidor, una herramienta hace el análisis de los servidores y la otra herramienta analiza la robustez de las contraseñas de los usuarios para evitar cualquier tipo de ataque.

Referencias

[1] Rea Guamán, M., Calvo-Manzano Villalón, J. A., & San Feliu Gilabert, T. (2018). Prototipo para gestionar la ciberseguridad en pequeñas empresas= A prototype to manage cybersecurity in small companies.

[2] Betancourt, C. E. A. (2017). Ciberseguridad en los sistemas de información de las universidades. Dominio de las Ciencias, 3(3), 200-217.

[3] Caneda Martínez, F. Ramas de la ciberseguridad: divisiones de una profesión con futuro. Revista: Campus Training. [Online]. Available: https://www. campustraining. es/noticias/ramas-ciberseguridad-profesion-futuro.

[4] Vargas Gutiérrez, G. (2021). Cómputo forense bajo Linux (Bachelor's thesis, Benemérita Universidad Autónoma de Puebla).

[5] Useche Lozano, C. A. (2015). Hacking ético, detección de vulnerabilidades en sistemas informáticos (Bachelor's thesis, Universidad Piloto de Colombia).

[6] Medina Rojas, E. F. (2015). Hacking Ético: Una herramienta para la seguridad informática (Bachelor's thesis, Universidad Piloto de Colombia).

[7] García Pérez, K. A. (2021). Aplicación de hacking ético mediante test de intrusión Pentesting para la detección y análisis de vulnerabilidades en la red inalámbrica de una institución educativa de la provincia de Santa Elena (Bachelor's thesis, La Libertad: Universidad Estatal Península de Santa Elena, 2021).

[8] Álvarez Roldán, M. Á., & Montoya Vargas, H. F. (2020). Ciberseguridad en las redes móviles de telecomunicaciones y su gestión de riesgos. Ingeniería y Desarrollo, 38(2), 279-297.

[9] Saucedo, A. L. H., & Miranda, J. M. (2015). Guía de ataques, vulnerabilidades, técnicas y herramientas para aplicaciones web. ReCIBE. Revista electrónica de Computación, Informática, Biomédica y Electrónica, (1).

[10] Ordóñez, V. D. C. Análisis Sobre Nikto Como Herramienta De Escaneo De Vulnerabilidades En Servidores Web.

[11] Gonzáles de Canales Gonzáles, E. (2014, 14 enero). Generación de reportes de vulnerabilidades y amenazas para aplicaciones web. [Online]. Available: http://hdl.handle.net/10609/26821

[12] Rawat, S., Bhatia, T., & Chopra, E. (2020). Web Application Vulnerability Exploitation using Penetration Testing scripts. Int. J. Sci. Res. Eng. Trends, 6(1), 311-317.

[13] Varghese, S., & Kurian, R. (2021). Identifying Vulnerabilities in a Website Using Uniscan and Comparing Uniscan, Grabber, Nikto. In Proceedings of the National Conference on Emerging Computer Applications (NCECA) (p. 225).

[14] Aguas, L., & Paredes-Parada, W. (2021). Aplicación de Vega Vulnerability Scanner en Aplicaciones Web. NEXOS CIENTÍFICOS-ISSN 2773-7489, 5(1), 1-8.

[15] Babincev, I. M., & Vuletić, D. V. (2016). Web application security analysis using the Kali Linux operating system. Vojnotehnički glasnik, 64(2), 513-531.

[16] Evwiekpaefe, A. E., & Habila, I. (2021). Implementing SQL Injection Vulnerability Assessment of an E-commerce Web Application using Vega and Nikto Tools.

[17] Mejía-Granda, C. M. (2018). Desarrollo de servicios web REST “inseguros” para auto-aprendizaje en la explotación de vulnerabilidades (Master's thesis).

[18] Mena Chuquimia, O. B. E. D. (2020). Técnicas de Seguridad Informática para Reducir las Vulnerabilidades por Inyección SQL en Aplicación Web.

[19] Hernández Mechate, E. J. (2020). Vulnerabilidades informáticas en el portal web de la Universidad Andina del Cusco.

[20] Sierra Huertas, T. La seguridad informática en el desarrollo de aplicaciones web mediante el uso de la metodología OWASP.

[21] Panchi Herrera, R. A. (2017). Estrategia para la detección de vulnerabilidades en la aplicación web de la Agencia Nacional de Tránsito como herramienta para la toma de decisiones (Master's thesis, Pontificia Universidad Católica del Ecuador).

[22] Bravo Sánchez, M. V., & Sánchez Prieto, D. A. (2018). Análisis de las amenazas, riesgos y vulnerabilidades del portal web del colegio católico José Engling mediante hackeo ético para el diseño y desarrollo de un aplicativo web de monitoreo de incidencias (Bachelor's thesis).

[23] Pérez González, R. C. (2016). Los sistemas de seguridad perimetral y principales vectores de ataque web.
[24] Barragán Montero, I. Recopilación de información para test de penetración.

[25] Martínez Salmerón, G. (2021). Herramientas para la Ruptura del Secreto de Contraseñas.

[26] Góngora Benítez, S. (2023). Realización de test de intrusión en sistemas informáticos.

[27] Muñoz Gallego, A. F. (2012). Guía práctica para el manejo de herramientas de seguridad para laboratorios docentes.

[28] Albarracín Lazo, C. A. (2011). Estudio de la Seguridad Informática y sus aplicaciones para prevenir la infiltración de los Hackers en las empresas (Bachelor's thesis, Quito: Universidad Israel, 2011).

[29]. Castillo Mendoza, J. I. (2022). Análisis de los sistemas de detección de intrusos (IDS) Open Source y Software Propietario (Bachelor's thesis, Babahoyo: UTB-FAFI. 2022).

[30] Cifuentes, J. H. Manual de Detección de Vulnerabilidades de Sistemas Operativ OS Linux Y Unix En Redes TCP/IP.

[31] Martínez Salmerón, G. (2021). Herramientas para la Ruptura del Secreto de Contraseñas.
, 2004.

Explorando efectos de los ransomwares en sistemas informáticos: acciones intrusivas, archivos encriptados y consecuencias devastadoras

Exploring the effects of ransomware on computer systems: Intrusive actions, encrypted files and devastating consequences

Juan Carlos Mejia Arguello, Ana Claudia Zenteno Vázquez, María del Carmen Santiago Díaz, Judith Pérez Marcial, Yeiny Romero Hernández, Gustavo Trinidad Rubín Linares
Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, Avenida San Claudio y 14 Sur, Ciudad Universitaria, Puebla, Puebla, C.P. 72570. México.
juan.mejiaar@alumno.buap.mx, {ana.zenteno, marucarmen.santiago, judith.perez, yeiny.romero, gustavo.rubin}@correo.buap.mx.

Abstract

To analyze the infection methods and encryption mechanisms used by modern ransomwares. It seeks to understand how these computer viruses manage to infect systems and files, and how they use cryptographic algorithms and encryption keys to block access to them. Detailed analysis of these aspects can help develop prevention and mitigation measures to reduce the impact of attacks.

Resumen

Analizar los métodos de infección y los mecanismos de cifrado que utilizan los ransomwares modernos. En particular, se busca comprender cómo estos virus informáticos logran infectar sistemas y archivos, y cómo usan algoritmos criptográficos y claves de cifrado para bloquear el acceso a los mismos. El análisis de estos aspectos puede ayudar a comprender y desarrollar medidas de prevención y mitigación para reducir el impacto de los ataques.

Keywords and phrases: Métodos de Infección, Mecanismos de Cifrado, Ransomware, Ataques, Bloqueo de Acceso.

1 Introducción

En la era digital en la que vivimos, la ciberseguridad es una preocupación cada vez más importante. Los ransomwares modernos son una de las amenazas más peligrosas para los usuarios de internet y empresas. Estos programas maliciosos utilizan diversos métodos de infección para infiltrarse en los sistemas de las víctimas, como correos electrónicos de phishing, descargas de software malicioso y vulnerabilidades de software no actualizado. Una vez que infectan un sistema, los ransomwares modernos utilizan algoritmos de cifrado avanzados para bloquear el acceso a los archivos y sistemas

de la víctima, exigiendo un rescate en forma de criptomonedas a cambio de una clave de descifrado. En esta introducción, exploraremos los métodos de infección y los mecanismos de cifrado utilizados por los ransomwares modernos, y cómo las víctimas pueden protegerse contra ellos [1].

Un ransomware es un tipo de software malicioso que se utiliza para bloquear el acceso a los archivos y sistemas de una víctima, exigiendo un rescate en forma de criptomonedas a cambio de una clave de descifrado. Este tipo de malware se propaga a través de diferentes canales, como correos electrónicos de phishing, descargas de software malicioso o explotación de vulnerabilidades en el software, y una vez que infecta un sistema, utiliza algoritmos de cifrado avanzados para cifrar los archivos de la víctima, impidiendo su acceso y uso [1]. El objetivo del ransomware es obligar a la víctima a pagar el rescate para recuperar el acceso a sus archivos y sistemas.

Ransomwares más conocidos:

- WannaCry: Es uno de los ransomwares más conocidos, propagado en 2017 a través de una vulnerabilidad en el protocolo SMB de Windows. Se extendió por todo el mundo, cifrando archivos y exigiendo un rescate.
- Petya/NotPetya: Apareció en 2016 y fue distribuido a través de una vulnerabilidad en el software de contabilidad ucraniano. Se ha utilizado principalmente para atacar grandes empresas y organismos gubernamentales.
- REvil/Sodinokibi: Es uno de los ransomwares más recientes y se ha utilizado principalmente para atacar empresas grandes y medianas. Utiliza un método de doble extorsión y se distribuye a través de kits de explotación de vulnerabilidades y correos electrónicos de phishing.
- GandCrab: Es uno de los ransomwares más prolíficos del 2018, distribuido a través de correos electrónicos de phishing y kits de explotación de vulnerabilidades. Utiliza un sistema de afiliados para distribuirse y se ha utilizado para atacar principalmente a pequeñas y medianas empresas.

Un exploit es una secuencia de comandos o un pequeño programa cuyo objetivo es aprovechar una vulnerabilidad de seguridad para conseguir un comportamiento no deseado del mismo, como un acceso no autorizado, toma de control del computador, entre otros. Puede tomar forma de un virus, un troyano o un script [1].

El protocolo SMB fue creado en 1985 y su función es la de compartir archivos, programas e impresoras, su número puerto es el 445. De la misma forma que otros protocolos pueden funcionar en modo cliente, servidor o ambos. EternalBlue es un exploit desarrollado por la Agencia de Seguridad Nacional de los Estados Unidos (NSA) como una herramienta de hacking para explotar una vulnerabilidad en el protocolo SMB de Windows. La vulnerabilidad fue filtrada en el 2017 por un grupo de hackers llamado "The Shadow Brokers", para posteriormente ser utilizada por este ransomware [7]. El exploit aprovecha una vulnerabilidad que acepta paquetes específicos de cualquier atacante, permitiéndoles ejecutar código en el escritorio remoto. Esto significa que este protocolo, además de solo enviar mensajes, archivos o lo que sea, podía ejecutarlos remotamente sin filtro.

Por otra parte, killswitch es un malware que es usado como una forma de controlar la propagación y el impacto del ransomware, implementa una forma de evitar ser detectados por investigadores de seguridad. En 2017. Se descubrió que el malware contenía un mecanismo de detección de una

dirección URL específica. Si el malware podía conectarse a esa dirección URL, se detenía automáticamente [7].

2 Marco teórico y estado del arte

Es importante destacar que los ransomwares pueden causar un daño significativo a las víctimas, incluyendo la pérdida de datos valiosos, interrupción de servicios y pérdida financiera. Aunque todos los virus son peligrosos, algunos pueden ser más destructivos que otros. Nos enfocaremos en dos ransomwares que han causado un gran impacto y daño en el pasado: WannaCry y NotPetya/Petya.

Según Mark Scott reconocido periodista del Times, en su artículo *"Los expertos buscan reducir los efectos secundarios del ciberataque WannaCry"*, varias organizaciones se han pronunciado sobre los efectos de este incidente, incluido el Ministerio del Interior de Rusia, la agencia de transporte y envío FedEx, así como el Servicio Nacional de Salud británico, NHS. En estos casos, los ciberdelincuentes tomaron el control, cifraron los datos y exigieron un rescate de $300 mil dólares para desbloquear los dispositivos. Estas instituciones se encuentran entre las más afectadas por este ataque [2].

La periodista Olivia Solon menciona en *"'Petya ransomware attack: ¿what is it and how can it be stopped?"*, que muchas organizaciones en Europa y EE. UU. se han visto paralizadas por un ataque de ransomware conocido como "Petya". El software malicioso se ha propagado a través de grandes empresas, la empresa de alimentos Mondelez, la firma legal DLA Piper y la empresa danesa de envío y transporte Maersk, lo que ha llevado a que las PC y los datos sean bloqueados y retenidos para pedir rescate [3].

Para dimensionar mejor la gravedad de lo ocurrido con Petya, Tomas Brewter, experto en ciberseguridad escribió para la revista Forbes en *"Petya Or NotPetya: Why The Latest Ransomware Is Deadlier Than WannaCry"* que considera que el mundo sufrió otra pesadilla de ransomware, con compañías farmacéuticas, sistemas de detección de radiación de Chernóbil, el metro de Kiev, un aeropuerto y bancos, todos afectados. Un hospital de EE. UU. también parece ser una víctima. Se espera lo peor, gracias a algunas características perniciosas en la muestra de ransomware [4].

2.1 Mecanismo de acción

Normalmente estos ransomwares se transmiten como un troyano o como un gusano infectando el sistema operativo, por ejemplo, con un archivo descargado o explotando una vulnerabilidad de software o incluso mediante un exploit.

WannaCry, llamado también mediante su nombre clave WannaCryptor 2.0 o mediante las diferentes formas que toma cuando se es infectado por el Ramsom: *Win32/WannaCrypt, TRJ/RamsomCriptk, Win32/Filecoder.Wannacryptor*, tiene la capacidad de infectar sistemas que van desde Windows XP hasta Windows 10, pasando por distribuciones hechas para servidores como lo son: Windows Server 2003, 2005 y 2012 [7].

Este virus usa principalmente el protocolo SMB (Server Message Block) como una forma de pasar a través de redes locales y en línea. También utiliza una vulnerabilidad en el protocolo SMB llamada

"EternalBlue" para propagarse a través de redes que no han aplicado los parches de seguridad necesarios para solucionar dicha vulnerabilidad, como se observa en la figura 1 y figura 2.

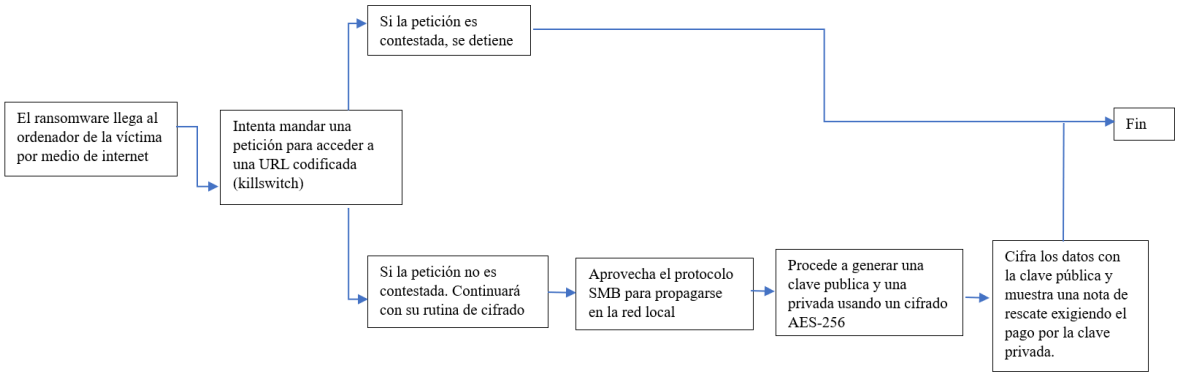


Figura 1. Diagrama de flujo con las acciones del ransomware.

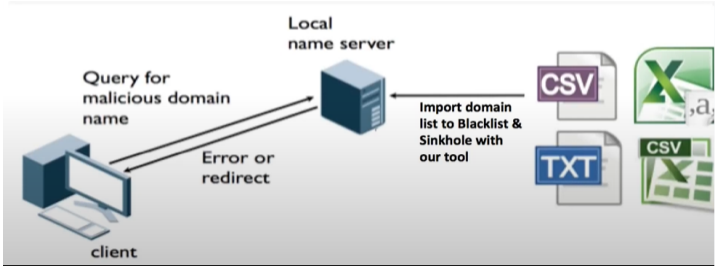


Figura 2. El ransomware envía solicitudes a un dominio anónimo. En caso de no se recibir respuesta automática, el software procedía a cifrar los archivos del sistema.

Este ransomware tuvo su origen en la utilización del método de phishing, en el que se enviaron cientos de correos electrónicos con suplantación de identidad. Tan solo con hacer clic en el correo electrónico, el malware se descargaba automáticamente en segundo plano. Al ejecutarse, este malware generaba los siguientes archivos:

- `¡WANNADECRYPTOR!.exe`: Este ejecutable asume la responsabilidad de presentar la información referente al rescate y llevar a cabo el proceso de descifrado de ciertos archivos. A partir de esta operación, se generan dos procesos en el administrador de tareas:
  - `MSSECSVE.exe`: Es el encargado de mostrar el programa de rescate
  - `TASKSCHE.exe`: Es el proceso que encripta los archivos
- `¡WANNADECRYPTOR!.exe.ink`
- `¡WANNADECRYPTOR!.BMP`
- `PlEASEREADME!.txt`

Además, creaba archivos temporales aleatorios para evitar su desinstalación; normalmente los creaba en las carpetas *roaming data* y *local temp*. Su propósito es eliminar la copia de seguridad almacenada en el sistema informático y así evitar la necesidad de restaurar el sistema si el archivo original está encriptado o dañado de alguna manera. Este proceso se implementó como una medida de seguridad proactiva para proteger la integridad de los datos y mitigar el impacto de posibles ataques cibernéticos, como el ransomware. Al eliminar las copias de seguridad almacenadas, se evitaba que los archivos de respaldo también fueran afectados por el cifrado malicioso [1].

Luego de ejecutar el comando con permisos de administrador, el virus continuó con su rutina de cifrado con cualquier archivo existente en el disco duro. No importa la extensión que tuvieran los archivos, este software finalmente los encripta y manda el mensaje que se observa en la figura 3.

Figura 3. Mensaje de pago de rescate.

Al mismo tiempo empezaba a aplicar los exploits que estaban en su código, el primer paso del virus es verificar si la computadora está en una red local o en una red compartida. Para ello utiliza la herramienta *DoblePulsar*, que le permitió obtener información sobre la configuración de la red del sistema. Si se determina que la computadora está conectada a una red local o compartida, el atacante pasa al siguiente paso en su estrategia.

Si se confirma la presencia de una red local o compartida, los atacantes explotan una lasitud especial llamada *EternalBlue* la cual aprovecha una vulnerabilidad en el protocolo de bloque de mensajes de servidor (SMB) de Windows conocido como MS17-010 [7]. Ejecutar malware en otras computadoras mediante el uso de EternalBlue permite a los atacantes aumentar la superficie de ataque y otorga más control sobre la infraestructura infectada. Cabe señalar que el uso de EternalBlue se basa en una vulnerabilidad descubierta y reparada por Microsoft en marzo de 2017 [9], pero muchos sistemas no han sido parcheados con las correcciones de seguridad correspondientes, lo que deja mucho espacio para que los atacantes exploten esta debilidad.

Así fue como el virus se propagó a nivel mundial; era suficiente con conectarse a una red Wi-Fi para que el ransomware se propagara con rapidez, y en caso de no pagar dentro del plazo estipulado, el programa desaparecía, dejando los archivos encriptados de manera irreversible como se observa en la figura 4. WannaCry usa un algoritmo de cifrado AES (Advanced Encryption Standard) de 128 bits para cifrar los archivos de los sistemas infectados. Este algoritmo es un estándar de cifrado altamente seguro y ampliamente utilizado en todo el mundo para proteger datos confidenciales [1]. El ransomware también utilizó un algoritmo de generación de claves llamado RSA (Rivest-Shamir-Adleman) para generar una clave de cifrado única para cada archivo cifrado, lo que dificultó aún más la recuperación de los datos sin la clave de descifrado correspondiente. Así que es

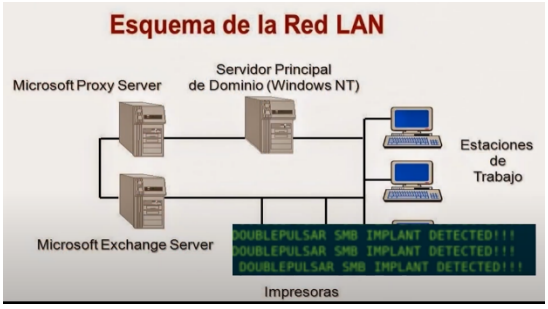


Figura 4. El exploit "DoublePulsar irrumpe en la conexión con servidores o estaciones de trabajo para la inyección de código y lograr la ejecución remota de código en el sistema comprometido.

3 El siguiente nivel de los ransomwares

Petya, una variante de malware que se ha utilizado en varios ataques devastadores en los últimos años. Este nuevo malware afecto a las computadoras justo después de los ataques de WannaCry, su nombre real es RAMSOM.PETYA también conocido como como WIN32/DISKODER.PETYA.A, TROJAN-RANSOM.WIN32.PTR.C. Su método de infección es vía internet, pero también puede hacerlo vía USB e infecta todos los sistemas operativos desde Windows XP, hasta Windows 10 [5].

Este ransomware fue descubierto y reportado en marzo del 2016 por la alemana Heise Security, especializada en seguridad informática y editorial de noticias tecnológicas, que se enfoca en temas relacionados con la seguridad en internet y la protección contra el cibercrimen [6]. Específicamente, Petya se aprovecha de la funcionalidad de páginas o programas que funcionan como sistemas de archivos [3], como lo es Dropbox, para infectar y propagarse por los dispositivos conectados a dicha red.

3.1 Método de infección más peligroso

El principal objetivo de este malware es infectar el MBR *(Master Boot Record)* para evitar cualquier acción del usuario. La razón por la que Petya ataca específicamente el MBR es porque este sector es extremadamente difícil de recuperar una vez que se ha visto comprometido, lo que hace que el virus sea especialmente peligroso y difícil de combatir.

El MBR es el primer sector de un dispositivo de almacenamiento de datos como el disco duro, conteniendo en él, un código de arranque.

Piso 1: La BIOS (Basic Input/Output System) o UEFI (Unified Extensible Firmware Interface) es el firmware que se ejecuta al inicio de la computadora y realiza una serie de comprobaciones de hardware.
Piso 2: La carga del bootloader (cargador de arranque) es el siguiente paso en el proceso de arranque. Es responsable de cargar el sistema operativo desde el disco duro o la unidad flash USB.
Piso 3: El MBR es la primera sección de un disco duro que se lee durante el proceso de arranque y contiene información sobre cómo se divide el disco en particiones.
Piso 4: El sistema operativo. En este punto, se cargan los controladores de dispositivo necesarios para que el sistema operativo pueda comunicarse con el hardware de la computadora.

Piso 5: Las aplicaciones y programas que se ejecutan, en primer plano.

Este virus realiza una acción destructiva al infectar y cifrar la tabla de particiones al romper físicamente la conexión entre el arranque del sistema operativo y el MBR (Master Boot Record). El sistema operativo requiere esta información para acceder y administrar los archivos almacenados en cada partición. Sin embargo, cuando el virus infecta y cifra la tabla de particiones, existe una clara diferencia entre el arranque del sistema operativo y el MBR. Es importante que el sistema operativo arranque correctamente y tenga acceso a las particiones y archivos correctos. Sin una conexión adecuada entre el MBR y la tabla de particiones, el sistema operativo no puede arrancar y funcionar. La acción del virus literalmente cruzó la línea entre el arranque del sistema operativo y el MBR lo cual tiene graves consecuencias [5].

3.2 Proceso de cifrado

Una vez infectado de este malware, al intentar iniciar o reiniciar la computadora aparecerá el mensaje de la figura 5. El verdadero mensaje que advierte de problemas en el disco duro realmente no está escrito con mayúsculas, así que este falso mensaje solo es una distracción, mientras el virus ya se encuentra cifrando todos los archivos.

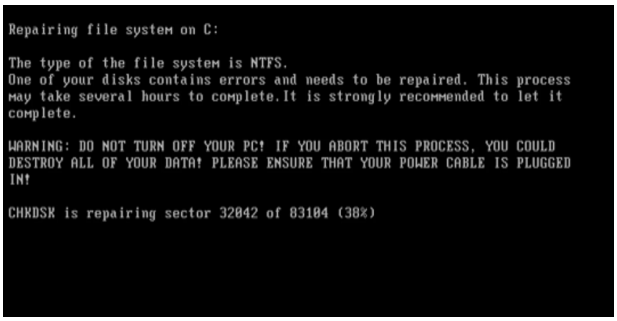


Figura 5. Mensaje falso del comando Check Disk.

La peculiaridad de este malware es que, una vez que ha cifrado los archivos del sistema, no exige la obtención de una clave o llave de desencriptación para poder acceder nuevamente a los datos, ya que Petya no tiene la intención de devolver los archivos al usuario, sino que los indicios sugieren que este malware está asociado con actividades respaldadas por un estado, específicamente vinculado a Rusia. Este enfoque altamente dirigido y geográficamente focalizado parece estar dirigido principalmente hacia Ucrania, lo que refuerza aún más la naturaleza política y de conflicto de este ataque. [8]. El hecho de que Petya no busque obtener un rescate, sino que actúe más bien como una herramienta de sabotaje digital, sugiere una nueva dimensión en la estrategia de ciberataques respaldados por naciones.

4 Análisis del procesos de acción del virus

Al ejecutarse, lo primero que hace es ingresar al host por medio de una herramienta conocida como *psexeccsvc.exe* la cual es un archivo que forma parte del software de Sysinternals PsExec, desarrollado por Microsoft. Este proceso se encarga de ejecutar otro proceso remotamente.

En caso de que por algún motivo falle dicho proceso el virus intenta acceder a la víctima con otro llamado *wniprvse.exe* el cual sirve para administrar servicios clientes en el sistema operativo y se activa automáticamente cuando la aplicación del cliente se conecta, principalmente se utiliza para monitorear los recursos del sistema.

Este virus aprovecha dicho proceso, no solo para ingresar, sino que también para controlar los procesos de servicios, así como otros archivos, y además de esto Petya también puede utilizar uno de estos dos procesos para copiarse a sí misma en tal caso de que el computador sea parte de una red compartida.

Después, se aloja en el sistema con el nombre PERFC.DAT a través de ese archivo se apodera del *rundll32.exe* el cual es un archivo utilizado por el sistema operativo para ejecutar automáticamente la funcionalidad de DLL, lo que permite a los desarrolladores de software implementar funciones reutilizables en sus aplicaciones sin tener que escribir el código completo, para ejecutar y llevar a cabo su rutina de cifrado de archivos. Este virus antes de cifrar los archivos primero extrae información importante del sistema infectado, para eso utiliza una herramienta llamada *Mimikatz*, la cual es utilizada para recuperar contraseñas y credenciales almacenadas en sistemas Windows y es de código abierto.

Mimikatz tiene la capacidad de extraer contraseñas y otros datos de autenticación almacenados en la memoria de un sistema operativo de Windows, incluyendo contraseñas de cuentas de usuario, contraseñas de servicios, contraseñas de red, claves privadas de certificados y otros datos relacionados con la autenticación.

Luego de extraer la información necesaria del computador el ransomware procede al cifrado de datos, primero verifica que tipo de antivirus utiliza la victima o si utiliza un antivirus si la condición dicha anteriormente es cierta, es decir hay un antivirus en la PC, se escribirá el código del malware en el MBR, lo que hará que el sistema no se pueda iniciar. Esto es para evitar una reacción inmediata del antivirus y evitar que lo elimine. Sin embargo, a pesar de que el ransomware sobrescribe el MBR, no encripta la MFT o tabla maestra de archivos que se observa en la figura 6. Esta tabla se encarga de almacenar como metadatos toda la información de los archivos, directorios, meta archivos, así como el nombre de archivo, fecha de creación, permisos de acceso y tamaño.

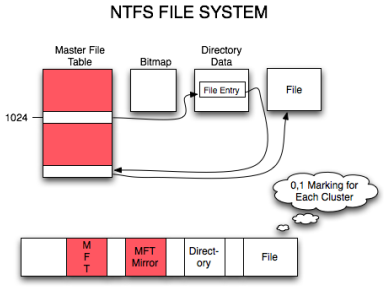


Figura 6. La MFT (Master File Table) es una base de datos interna utilizada por el sistema de archivos NTFS de Windows para mantener un registro de todos los archivos y directorios almacenados en una unidad de disco.

Al no encriptar esta tabla, surge la posibilidad de recuperar los datos esto es en caso de que el usuario haya hecho una copia de seguridad de su MBR si lo hizo entonces tranquilamente puede restaurar el sistema. En caso de no contar con antivirus Petya procede a sobrescribir los 32 sectores de almacenamiento del disco duro, desde el sector 0 hasta el sector 18 e incluirá su propio programa de arranque o sea el falso Check Disk que se observa en la figura 5.

Sobrescribe el sector 32 en donde se hallan los datos estructurados, los cuales contienen el cifrado del MFT y el sector 34 en donde se aloja el MBR encriptado, los derechos de administrador son necesarios para escribir en el MBR, si se realiza una ejecución directa del malware sin derechos de administrador no se escribirá en él, sin embargo, dado a su forma de ingresar al host este virus adquiere tanto derechos de usuario como derechos de administrador.

5 Conclusiones

Se recomienda mantener actualizado el sistema operativo y contar con un programa antivirus confiable que es sumamente importante en cualquier situación de seguridad informática. Estas precauciones son necesarias para proteger el sistema y reducir el riesgo de varios tipos de ataques, incluidos aquellos que pueden infectar gravemente el sistema. Actualizar su sistema operativo implica instalar parches y actualizaciones proporcionados por el proveedor del software, a menudo incluyen correcciones de seguridad que abordan vulnerabilidades conocidas y ayudan a proteger su sistema de amenazas potenciales. Mantener su sistema actualizado reducirá las posibilidades de ser explotado y aumentará su resistencia a los ataques.

Una solución robusta de monitoreo de red y firewall puede ayudar a detectar y bloquear el tráfico malicioso para que no ingrese a su sistema. Estas medidas de seguridad adicionales actúan como una barrera protectora que evita que el ransomware se infiltre en su red y comprometa sus datos. En cuanto a los antivirus, es importante tener en cuenta que no todos los antivirus son iguales, y algunos pueden ser más efectivos que otros en la detección y eliminación de virus específicos. Sin embargo, en el caso de un virus grave y desconocido, es difícil recomendar un antivirus en particular, ya que la efectividad de cualquier antivirus puede variar dependiendo del virus específico y de cómo se haya diseñado para evadir la detección. Por último, tener cuidado con los correos electrónicos y enlaces sospechosos, los correos electrónicos y enlaces maliciosos son uno de los principales vectores de ataque del ransomware. No abrir correos electrónicos de remitentes desconocidos, no hacer clic en enlaces ni descargar archivos adjuntos sospechosos. Así como tener cuidado al interactuar con contenido de fuentes no confiables.

Referencias

[1] A brief study of Wannacry Threat: Ransomware Attack 2017. International Journal of Advanced. Research in Computer Science. Available: https://sbgsmedia.in/2018/05/10/2261f190e292ad93d6887198d7050dec.pdf [Accessed: 10/Mayo/2023].

[2]Scott, M. A. Los expertos buscan reducir los efectos secundarios del ciberataque “WannaCry”. The New York Times. Available: https://www.nytimes.com/es/2017/05/13/espanol/efectos-ciberataque-wannacry-ransomware.html [Accessed: 17/Abril/2023].

[3] Solon. O. A 'Petya' ransomware attack: what is it and how can it be stopped? The Guardian. Available: https://www.theguardian.com/technology/2017/jun/27/petya-ransomwarecyber-attack-who-what-why-how [Accessed: 17/Abril/2023].
[4] Brewster, T. A.Petya Or NotPetya: Why The Latest Ransomware Is Deadlier Than WannaCry. Forbes. Available: https://www.forbes.com/sites/thomasbrewster/2017/06/27/petya-notpetya-ransomware-is-more-powerful-than-wannacry/?sh=74a49f42532e [Accessed: 17/Abril/2023].

[5] Ren, A. L. Y., Liang, C., Hyug, I. J., Broh, S. N., & Zaman, N. A Three-Level Ransomware Detection and Prevention Mechanism. EAI Endorsed Transactions on Energy Web, 0(0), 162691. https://doi.org/10.4108/eai.13-7-2018.162691 [Accessed: 17/Abril/2023].

[6]Belcic, I. ¿Qué es el ransomware Petya y por qué es tan peligroso? Available: https://www.avast.com/eses/cpetya. [Accessed: 17/Abril/2023].

[7] The dynamic analysis of WannaCry ransomware. (2018, 1 febrero). IEEE Conference Publication | IEEE Xplore. Available: https://ieeexplore.ieee.org/abstract/document/8323682 [Accessed: 17/Abril/2023].

[8] Ucrania asegura tener pruebas que involucran a Rusia en el ciberataque Petya. IDG Communications S.A.U. Available : https://cso.computerworld.es/alertas/ucrania-asegura-tener-pruebas-que-involucran-arusia-en-el ciberataque-petya [Accessed: 10/Mayo/2023].

[9] WannaCry: tres años después sigue siendo una amenaza activa de la cual debemos aprender | WeLiveSecurity. (2023, April 4). WeLiveSecurity. Available: https://www.welivesecurity.com/la-es/2020/05/12/wannacry-tres-anos-despues-una-amenaza-activa-debemos-aprender/ [Accessed: 10/Mayo/2023].

Innovaciones de las Ciencias Computacionales en Sistemas Inteligentes y Ciberseguridad
se terminó de editar en Diciembre de 2023 en la
Facultad de Ciencias de la Computación
Av. San Claudio y 14 Sur Jardines de San Manuel
Ciudad Universitaria
C.P. 72570

Innovaciones de las Ciencias Computacionales en Sistemas Inteligentes y Ciberseguridad

Coordinado por María del Carmen Santiago Díaz