

Prueba de autenticación para el logueo web mediante algoritmo genético



Colaboración

Fernando Alberto Hernández Guevara; Keila Elena Ocaña Drouaillet; Roberto Ángel Meléndez Armenta; Jorge Mario Figueroa García, Instituto Tecnológico Superior de Misantla

RESUMEN: Los algoritmos evolutivos tienen su aplicación en la ingeniería en problemas complejos de optimización, las pruebas de Turing se basan en la interrogación para diferenciar computadoras de humanos, derivado de que los sistemas web por su naturaleza pública y multiplataforma son susceptibles a vulnerabilidades frente a ciber atacantes, implementar una prueba de turing automática y pública, con algoritmos genéticos, para diferenciar computadoras de humanos y así facilitar el acceso de los usuarios en el Sistema Integral del Tecnológico de Misantla (SITM), aunque en el mercado hay tecnologías para el control de acceso, ninguna cumple con las necesidades primordiales para minimizar las posibles amenazas a la seguridad del SITM. En su desarrollo se usó el paradigma de la POO, con la finalidad de construir una función objetivo con “ciertas regularidades”, es decir que verifiquen que sus respectivos valores en las funciones sean similares. Logrando con ello la creación de un algoritmo genético que asegura la identificación, autenticación y autorización a aquella información que se almacena, procesa y comunica, elevando la seguridad del Sistema Integral del Tecnológico de Misantla.

PALABRAS CLAVE: captcha, algoritmo genético, backend, frontend.

ABSTRACT: Evolutionary algorithms have their application in engineering in complex optimization problems, Turing tests rely on interrogation to differentiate computers from humans, derived from the fact that web systems due to their public and cross-platform nature are susceptible to vulnerabilities against cyber attackers, implement an automatic and public turing test, with genetic algorithms, to differentiate computers from humans and thus facilitate user access in the Misantla Technological Integral System (SITM), Although there are technologies for access control on the market, none of them meet the essential needs to minimize possible threats to SITM security. In its development, the OOP paradigm was used, in order to build an objective function with “certain regularities”, that is, verify that their respective values in the functions are similar. Thus achieving the creation of a genetic algorithm that ensures the identification, authentication and authorization of that information that is stored, processes and communicates, increasing the security of the Misantla Technological Integral System.

KEYWORDS: captcha, Genetic algorithm, backend, frontend.

INTRODUCCIÓN

Desde el principio de los tiempos del surgimiento del término informática el humano ha tratado de replicar las operaciones rutinarias que realiza de forma manual o análoga, empezando desde pequeños problemas matemáticos hasta el nivel biológico, como son la inteligencia artificial, la cual no es lo mismo que replicar el pensamiento humano en una computadora o una tecnología que remplace lo biológico a lo digital, surgiendo con esto la inteligencia artificial, que es el campo científico de la informática que se centra en la creación de programas y mecanismos que pueden mostrar comportamientos considerados inteligentes, esto puede resumirse como la capacidad de adaptación al entorno para resolver los problemas que se le presenten, lo que nos permite implementar en las máquinas un mecanismo para simular el comportamiento de un ser vivo, capaz de

4. ECMAScript: es una especificación de lenguaje de programación publicada por ECMA International. ECMAScript define un lenguaje de tipos dinámicos ligeramente inspirado en Java y otros lenguajes del estilo de C. Soporta algunas características de la programación orientada a objetos mediante objetos basados en prototipos y pseudoclases[5].

5. JavaScript: es una importante tecnología web que provee de interactividad y efectos especiales las páginas web. Cada vez que una página web hace más que estar ahí y mostrar información estática para que puedas ver mostrando actualizaciones de contenido oportunas, o mapas interactivos, o gráficos animados 2D/3D, o desplazarse por reproductores de video, etc. puedes apostar a que JavaScript probablemente esté involucrado[6].

6. JSON: es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del lenguaje de programación JavaScript, standard ECMA-262.[7].

RESULTADOS

Algoritmo Genético

Los algoritmos genéticos que simulan la evolución utilizando el código de computadora para ayudar a encontrar soluciones óptimas cuando se enfrentan problemas que involucran múltiples requisitos, son parte de una clase de algoritmos evolutivos que conforman un segmento más grande, el software de aprendizaje automático.

Como son la integración de una prueba de turing con algoritmos genéticos (AG), para la identificación entre un humano y un algoritmo mal intencionado (spam), que busca el acceso a un sistema de información mediante la autenticación a través de un formulario.

Los algoritmos evolutivos tienen su aplicación en la ingeniería en problemas complejos de optimización. De manera particular en esta investigación se utiliza un AG con la estructura de una prueba de turing, con el fin de diferenciar computadoras de humanos.

El funcionamiento del algoritmo genético parte de una población inicial, es definida por un conjunto de individuos. Cada individuo de la población es una propuesta de solución al problema.

En este caso de estudio, la población inicial se definió aleatoriamente. Cada individuo es evaluado mediante una función objetivo con el propósito de evaluar la aptitud de este, para después pasar al proceso de selección.

Una vez seleccionado un par de individuos, se cruzan para generar dos nuevos individuos. Tras el cruce se da lugar a la mutación, consiste en modificar a uno o varios de sus genes de manera aleatoria. Para el criterio de convergencia o terminación, en este caso se limitó a un número máximo de evoluciones (generaciones).

El pseudocódigo de un algoritmo evolutivo, se expresa de la siguiente forma:

```
BEGIN /* Algoritmo evolutivo Simple */
  Generar una población inicial
  Computar la función de evaluación de cada individuo.
  FOR Generaciones DO
    BEGIN /* Producir nueva generación */
      FOR Sobrevivientes DO
        BEGIN
          Seleccionar aleatoriamente individuo sobreviviente.
          Mutar sobreviviente con cierta probabilidad.
          Insertar sobreviviente en la nueva población.
        END
      FOR Poblaciontotal-Sobrevivientes DO
        BEGIN /* Ciclo Reproductivo */
          Seleccionar dos individuos de la anterior generación, para el cruce (Selección por ruleta).
          Cruzar con cierta probabilidad los dos individuos obteniendo dos descendientes.
          Mutar descendiente con cierta probabilidad.
          Computar la función de evaluación del descendiente mutado.
          Insertar descendiente mutado en la nueva generación.
        END
      END
    END
  END
```

Codificación de los genes: el primer paso es determinar cómo representar los datos, esta es la base de la AG, que representa al individuo a través de genes que imitan algún elemento de los datos, para esto se representaron los genes como caracteres (caracteres ASCII), cada carácter puede provenir de uno de estas 26 letras: ABCDEFGHIJKLM-NOPQRSTUVWXYZ. A su vez se necesita un objetivo, en este caso, es la palabra que el usuario ingrese.

Población inicial: la población inicial parte de la inicialización del individuo que consiste en pre-asignar los valores de los caracteres en la codificación de los genes, es principalmente una clase de conveniencia que mantiene a todos nuestros individuos unidos como una unidad.

Selección: para garantizar que los mejores individuos tienen una mayor posibilidad de ser padres, frente a los individuos menos buenos. se debe de tener precaución para dar una oportunidad de reproducirse a los individuos menos buenos. Éstos

pueden incluir material genético útil en el proceso de reproducción. Esta idea define la presión selectiva que determina en qué grado la reproducción está dirigida por los mejores individuos. En la clase se usa un enfoque de selección de ruleta (Seleccionar por genes aleatorios y elegir el mejor).

Cruza: se puede tener uno o más operadores de cruce para la representación. Algunos aspectos importantes a tener en cuenta son:

- Los hijos deberían heredar algunas características de cada padre.
- Si éste no es el caso, entonces estamos ante un operador de mutación.
- Se debe diseñar de acuerdo a la representación.
- La recombinación debe producir cromosomas válidos.
- Se utiliza con una probabilidad alta de actuación sobre cada pareja de padres a cruzar, el punto de cruce puede ser entre 0.5 y 0.9, si no actúan los padres son los descendientes del proceso de recombinación de la pareja.

Para el experimento descrito en el presente el punto de cruce (tasa uniforme) fue determinado al 0.5%.

Mutación: al igual que en el cruce, en la mutación se puede tener uno o más operadores de mutación para nuestra representación. Pero existen aspectos a considerar para realizar el proceso:

- Debe permitir alcanzar cualquier parte del espacio de búsqueda.
 - El tamaño de la mutación debe ser controlado.
 - Debe producir cromosomas válidos.
 - Se aplica con una probabilidad muy baja de actuación sobre cada descendiente obtenido tras aplicar el operador de cruce.
- La tasa de valores aleatorios está mutada al 0.01 %.

Tabla 1. Resumen de los parámetros del algoritmo evolutivo.

Parámetro	Valor
Tamaño de la población.	Depende del objetivo.
Tamaño del Cromosoma	27
Selección	Ruleta
Porcentaje de cruce	0.5%
Probabilidad de mutación	0.1%
Numero de generaciones	200
Corridas	6
Métrica	Menor Costo

Aplicación

El ambiente de desarrollo del algoritmo evolutivo se realizó en un lenguaje de programación PHP. La funcionalidad del algoritmo evolutivo se rige mediante los parámetros descritos anteriormente y resumidos en la Tabla 1. Resumen de los parámetros del algoritmo evolutivo.

Después de que los parámetros del algoritmo estocástico están configurados, se procedió a realizar la

implementación, la cual se describe a continuación. La Figura 2. Integración AG, muestra la composición de un algoritmo genético y una prueba de turing, la cual le solicita al usuario la siguiente instrucción:

¿Eres un humano?

Forma una palabra con los siguientes caracteres: (ABCDEFGHIJKLMNPOQRSTUVWXYZ).

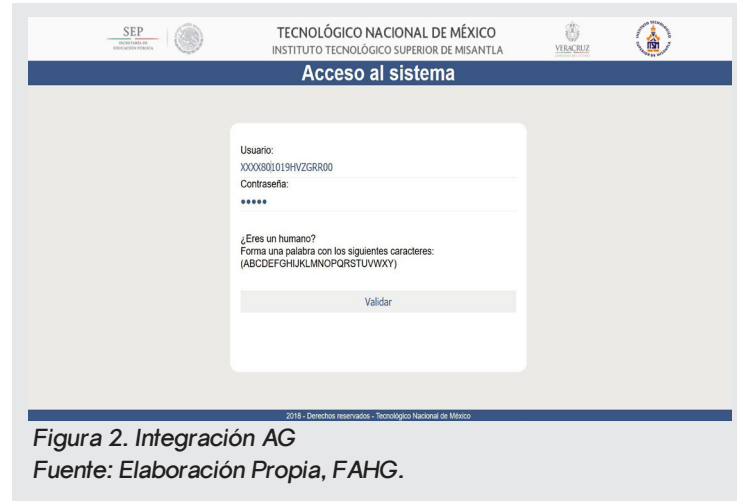


Figura 2. Integración AG

Fuente: Elaboración Propia, FAHG.

Posteriormente, se ingresa la palabra que el usuario decida siempre y cuando sea en mayúsculas y se elabore con las letras que le muestra en pantalla, se da clic en validar, para verificar que el humano está cumpliendo con la instrucción y no sea un algoritmo mal intencionado que requiere saturar el sistema con la solicitud de peticiones, como se puede observar en la Figura 3. Ingreso de la palabra en AG.

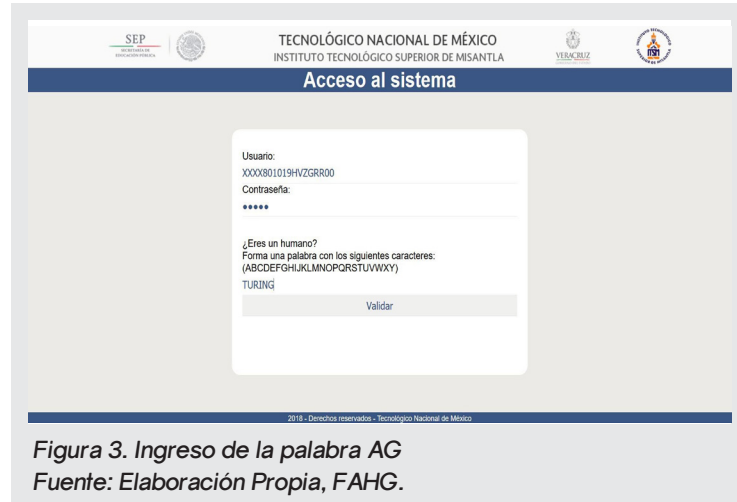


Figura 3. Ingreso de la palabra AG

Fuente: Elaboración Propia, FAHG.

Después de dar clic en la validación, se corre el algoritmo genético, el cual internamente tiene la regla que se le mostró al usuario, que le indica que formule una palabra con los siguientes caracteres "ABCDEFGHIJKLMNPOQRSTUVWXYZ", el AG vuelve la palabra tecleada por el usuario su objetivo, con base en esto debe de lograr la palabra ingresada a partir de dicha restricción, cabe mencionar que

la palabra puede ser cualquiera, con cualquier carácter conocido, pero para tales casos donde no se respete la norma no se alcanzará la meta, siempre y cuando se cumpla el criterio el AG llegará a su objetivo y se habilitará el botón desbloquear, como se puede observar en la Figura 4. Corrida AG.

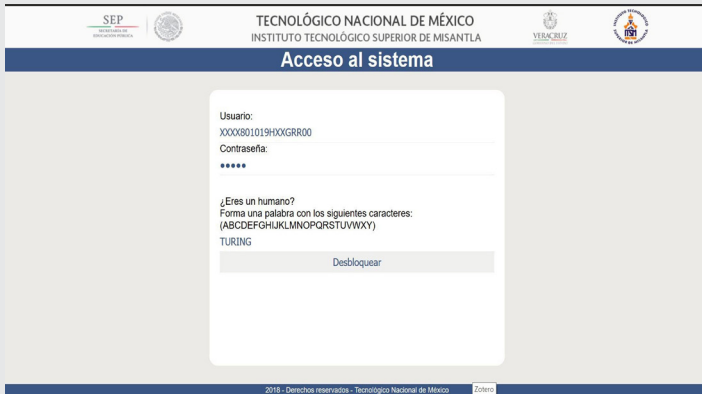


Figura 4. Corrida AG
Fuente: Elaboración Propia, FAHG.

En la Figura 5. Botón ingresar, se puede observar que después de dar clic en el botón desbloquear, si el AG llegó a su objetivo, se habilita el botón de ingresar y se accede a la pantalla principal.



Figura 5. Botón ingresar
Fuente: Elaboración Propia, FAHG.

En la Figura 6. Autenticación correcta, se puede visualizar la pantalla del sistema, después de cumplir los requisitos previos para el mismo.

En caso de que el usuario no comprenda la instrucción e ingrese algo erróneo o que el algoritmo genético no alcance el objetivo, el sistema enviará un mensaje de error, como se puede observar en la Figura 7. Error AG, posterior a dar clic en desbloquear, ya que es la parte del proceso donde se valida que el AG llegue al objetivo (la palabra que ingresa el usuario final).



Figura 6. Autenticación correcta
Fuente: Elaboración Propia, FAHG.



Figura 7. Error AG
Fuente: Elaboración Propia, FAHG.

Validación

En la tabla 2 se muestran una comparación con los resultados obtenidos al momento de ejecución de AG con distintos tamaños en el objetivo. En la columna tiempo, describe en segundos cuanto tardo el proceso de ejecución para la resolución de la instancia.

Tabla 2. Resultados obtenidos en las instancias del experimento.

Instancia	Matriz	Tamaño	Generaciones	Fitness	Tiempo
Instancia_1	1x4	4	9	1	.04s
Instancia_2	1x4	4	24	1	.01s
Instancia_3	1x4	4	29	1	0.013s
Instancia_4	1x11	11	167	1	.063s
Instancia_5	1x11	11	124	1	.06s
Instancia_6	1x11	11	45	3	.038s

En la comparativa que se muestra en la tabla 3, se describen los tiempos que se llevó en la resolución de 6 objetivos para el AG y la misma cantidad para el captcha, en la columna instancia, describe el número de objetivo a alcanzar y/o prueba realizada, igual que para captcha, en la columna tiempo, se visualiza el tiempo que tardo en cada uno de ellos para llegar a la solución correspondiente. Se puede ver que después de los experimentos realizados el AG tiene una ventaja para tales casos.

Tabla 3. Comparativa entre un captcha tradicional y la solución propuesta.

AG		CAPTCHA	
Instancia	Tiempo	Instancia	Tiempo
Objetivo1	.04s	Objetivo1	13s
Objetivo2	.01s	Objetivo2	3s
Objetivo3	0.013s	Objetivo3	2.45s
Objetivo4	.063s	Objetivo4	2.62s
Objetivo5	.06s	Objetivo5	2.62s
Objetivo6	.038s	Objetivo6	3.81s

[4] P. Prescott, *HTML 5*. Babelcube Inc., 2015.
 [5] N. Prusty, *Learning ECMAScript 6*. Packt Publishing Ltd, 2015.

[6] C. Vigouroux, *Aprender a desarrollar con JavaScript*. Ediciones ENI, 2017.

[7] A. Gangemi et al., *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*. Springer, 2018.

CONCLUSIONES

Como producto de esta investigación se logró realizar la integración de un algoritmo evolutivo con una prueba de Turing, enfocado a un ámbito web para el Sistema Integral del Tecnológico de Misantla y así garantizar que es un humano el que trata de ingresar y no un algoritmo automático.

Para obtener un funcionamiento óptimo del algoritmo genético, se recomienda un número mayor en el número de generaciones, pues entre más generaciones sean evaluadas se obtendrá un mejor resultado en el desempeño de los individuos.

Un problema habitual en las ejecuciones de los AG, surge debido a la alta velocidad con la que el algoritmo converge, en algunos casos la convergencia es muy rápida, lo que suele denominarse convergencia prematura, en la cual el algoritmo converge hacia óptimos locales, mientras que en otros casos el problema es justo el contrario, es decir se produce una convergencia lenta del AG. Al comparar la función objetivo del AG propuesto en el presente trabajo contra el CAPTCHA tradicional, la obtención de n funciones objetivo aproximadas puede resultar mejor que el captcha tradicional, en el supuesto caso que la evaluación aproximada resulta como mínimo n veces más rápida que el captcha y a su vez garantiza que quién inicia sesión en el Sistema Integral del Tecnológico de Misantla (SITM) es un humano o persona real y no un algoritmo de spam.

BIBLIOGRAFÍA

[1] X. G. GARCÍA, *Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión*. Ediciones Paraninfo, S.A., 2014.

[2] J. M. E. González, J. G. Caballero, y A. N. García, *Integración de sistemas de automatización industrial Edición 2019*. Ediciones Paraninfo, S.A., 2019.

[3] C. P. Millahual, *PHP 7 - Sitios Dinámicos: Aprenda a programar sin conocimientos previos*. RedUsers, 2019.