Una Propuesta de Reconocimiento de Patrones en el Tráfico de Red basada en Algoritmos Genéticos.

Carlos A. Catania, Carlos García Garino.

Laboratorio para la Producción Integral por Computadora - LAPIC
Instituto Tecnológico Universitario, Universidad Nacional de Cuyo
{ccatania,cgarcia}@itu.uncu.edu.ar

Resumen El reconocimiento de patrones en el tráfico de red es uno de los componentes fundamentales de los sistemas de detección de intrusos. En este trabajo se estudian las posibilidades de aplicación de un algoritmo genético para obtener reglas que permitan reconocer las instancias de tráfico normales. Este enfoque es distinto al propuesto en los trabajos anteriores, en donde se busca obtener los patrones de tráfico de las instancias que presentan anomalías. En el presente trabajo se discuten y proponen los ajustes necesarios a un algoritmo genético canónico, fundamentalmente en lo que se refiere a la función de fitness y a las técnicas para garantizar la convergencia hacia múltiples soluciones.

1. Introducción.

La seguridad de las redes de datos se ha transformado en un serio problema en los últimos años. El crecimiento vertiginoso que ha presentado la Internet ha permitido mostrar las fallas de seguridad en las implementaciones de los protocolos de red subyacentes. Situación que resulta comprensible, ya que muchos de estos protocolos originalmente fueron pensados para unir de 10 a 50 computadoras de las universidades de los Estados Unidos.

Las falencias en la seguridad de protocolos como *ARP*, *TCP*, *TELNET*, *SMTP*, *FTP* han sido la causa de ataques contra la confidencialidad, la disponibilidad y la autenticidad de los datos transportados. Si bien estos problemas han sido corregidos a lo largo de los años, continuamente se van descubriendo nuevas maneras de realizar estos ataques.

El ingeniero en seguridad de redes debe estar alerta para detectar estos ataques, informándose de las nuevas vulnerabilidades descubiertas o tipos de ataques perpetrados. Sin embargo una gran cantidad de estos ataques tienen lugar antes que se conozcan siquiera las vulnerabilidades o fallas que los provoca.

Para hacer frente a esto es que en los últimos años han surgido propuestas para la aplicación de técnicas de inteligencia artificial en el ámbito de la seguridad en redes.

En este trabajo se presenta una propuesta de un algoritmo genético para el reconocimiento de patrones en el tráfico de red como punto de partida para

abordar un problema de mayor envergadura como lo es la detección de intrusos por anomalías en el tráfico de red. Entendiéndose por una anomalía a toda instancia de tráfico que se aparte del comportamiento normal de la red [1]. El algoritmo propuesto parte de una población de individuos conformados por instancias de tráfico de red elegidas al azar, para obtener, al final del proceso, el conjunto de reglas que más coincidencias encuentre en el tráfico de la red.

El trabajo esta organizado de la siguiente manera: En la sección 2 se discuten los trabajos más relevantes relacionados con la aplicación de algoritmos genéticos al dominio. Posteriormente en la sección 3 partiendo de los trabajos comentados en los antecedentes se discuten y proponen los ajustes a un algoritmo genético general, necesarios para su aplicación en la búsqueda de patrones en el tráfico de red. Fundamentalmente se discuten, la codificación elegida para representar a la población de individuos y las funciones de optimización utilizadas. Se analizan también las técnicas utilizadas para garantizar la convergencia hacia múltiples soluciones como crowding y sharing. La sección 4 presenta los resultados obtenidos aplicando el algoritmo propuesto a un caso de estudio. Finalmente en la sección 5 se presentan las conclusiones y los trabajos futuros.

2. Antecedentes.

La utilización de algoritmos genéticos para la detección de intrusos ha sido estudiada con anterioridad [2,3,13,18,20].

Li [3]y Gong et al. [2] exploran la utilización de algoritmos genéticos para la generación de reglas de clasificación en el tráfico de red, para lo cual buscan los patrones más comunes en las instancias de tráfico anómalas. Para esto se necesita un conjunto de instancias de tráfico que previamente haya sido analizado por un experto en seguridad de redes, en el cual éste haya resaltado las instancias conteniendo anomalías. Una vez finalizado el entrenamiento, el algoritmo es capaz de generar alertas cuando observe alguno de los patrones aprendidos.

Sinclair et al. [18] propone la utilización del algoritmos genéticos para encontrar reglas sencillas que permitan encontrar patrones en el tráfico de la red combinado con arboles de decisión. Otros autores [4,13] utilizan programación genética para obtener la reglas de clasificación. Esta variante permite obtener reglas de mayor complejidad. Por otra parte en [12,14,15] se estudian diferentes aplicaciones de los algoritmos genéticos en el contexto de la detección de intrusos.

Se discuten a continuación los aspectos más relevantes de las implementaciones mencionadas.

2.1. Representación de los datos.

La manera de representar los individuos de la población es una de los principales componentes de los algoritmos genéticos que necesita ser adaptado al dominio de aplicación.

Los trabajos mencionados en los antecedentes destacan ciertos atributos de una conexión de red que resultan más relevantes para la detección de anomalías.

En [18] se consideran los atributos provenientes de la cabecera del paquete IP y del segmento TCP como: puerto origen, puerto destino, dirección IP origen y dirección IP destino. En otros trabajos [2,3] se tienen en cuenta además otros atributos como información sobre el tiempo de duración de la conexión y la cantidad de bytes recibidos y transferidos. En [3,4,18] se destaca la posibilidad de descartar de manera aleatoria algún gen dentro del cromosoma del individuo con el fin de obtener individuos más generales que puedan encontrar coincidencias con un mayor número de reglas.

2.2. Función de optimización.

Crosbie [20] propone una función de optimización que penaliza a los individuos en función de un ranking que indica el grado de dificultad presentado para la detección de esa instancia de tráfico. Li [3] adapta la propuesta anterior y propone la utilización de pesos para destacar ciertos atributos de las instancias que tráfico que son considerados más importantes en el dominio de la detección de intrusos.

En [2,4] se propone la utilización de una función de fitness basada en el *support-confidence framework* [19], en donde si una regla es descripta como *si* A *entonces* B se puede determinar el fitness de dicha regla siguiendo la ecuación:

$$\begin{aligned} support &= |A \cup B|/N \\ confidence &= |A \cup B|/|A| \\ fitness &= w_1 * support + w_2 * confidence \end{aligned}$$

Donde variando los valores de w_1 y $w_2$se pueden detectar las anomalías de manera general o también clasificar de manera precisa los distintos tipos de anomalías.

2.3. Otros operadores.

Para los operadores de mutación, cruzamiento y selección no se detallan implementaciones más allá de las propuestas de la literatura clásica de algoritmos genéticos [5].

2.4. Técnicas de nicho.

Como mencionan Miller y Shaw [6], los algoritmos genéticos utilizan poblaciones que con el tiempo convergen hacia una única mejor solución. Aplicado al dominio de este trabajo, el resultado estaría en aquel individuo que tenga coincidencias con la mayor cantidad instancias de tráfico. Dado que es altamente improbable que un solo individuo pueda ser un buen representante de todas las instancias de tráfico, resulta mucho más efectivo contar con un número mayor de individuos que presenten coincidencias con algunas instancias de tráfico.

En este contexto en la literatura [6,7,8] se mencionan técnicas de nicho, como crowding y sharing. Las mismas se basan en el concepto de proximidad, expresado mediante una función distancia, que permite encontrar a los individuos más cercanos. Crowding se basa en escoger unos pocos individuos al azar y reemplazar uno de los mismos por un nuevo individuo en base a criterios de proximidad [6]. Sharing en cambio degrada el fitness de un individuo en función de la cantidad de individuos que estén próximos al mismo.

En [3,18,19] se destaca la utilización de una variante de crowding aunque no se menciona cual, ni se dan detalles de su implementación. Sinclair et al en [18] plantea la utilización de una variante de crowding utilizando distancia de Hamming para encontrar los individuos más cercanos.

Lu [4] propone una técnica diferente basada en la competición por token. Cada instancia de tráfico del conjunto de entrenamiento contiene un token. Si un individuo coincide con la instancia de tráfico adquiere el token. La prioridad para obtener el token se basa en una probabilidad basada en la calidad de cada individuo. Finalmente la cantidad de tokens obtenidos por cada individuo es utilizado como un parámetro de la función de fitness.

3. Algoritmo genético propuesto.

Tomando como base los trabajos anteriores y un algoritmo genético canónico como el descripto por Goldberg [5] se presenta una propuesta de algoritmo genético para la búsqueda de patrones que permitan reconocer las instancias normales en el tráfico de red.

Este enfoque es distinto al propuesto en los trabajos mencionados en los antecedentes (ver sección 2), en donde se busca obtener los patrones de tráfico de las instancias que presentan anomalías. Sin embargo muchas de las técnicas utilizadas en el estado del arte se pueden emplear en el contexto del presente trabajo.

En las siguientes subsecciones se discuten la representación de la población, la función de optimización y las técnicas para mantener un conjunto de soluciones posibles, las cuales conforman el algoritmo propuesto. El algoritmo utiliza operadores de selección, cruzamiento y mutación, los cuales no presentan aportes respecto a lo mencionado en el estado del arte.

3.1. Representación de la población.

Para este trabajo se seleccionan 6 atributos de una instancia de tráfico: tiempo de duración de la conexión, tipo de protocolo, puerto origen, puerto destino, dirección IP origen y dirección IP destino.

Los atributos se representan como una lista de genes, de acuerdo a la estructura indicada en el cuadro 1. A continuación se muestra un ejemplo de atributo:

Ejemplo 1. (0,0,2,5,2114,80,192,168,1,1,192.168,1,2)

Cuadro 1. Estructura del cromosoma de un individuo.

Atributo	Nro. de genes	Valor máximo
HH:MM:SS	3	60
Puerto origen	1	65535
Puerto destino	1	65535
Dirección origen	4	255
Dirección destino	4	255

El ejemplo representa una conexión http de una duración de 2 segundos con puerto origen 2114, puerto destino 80, dirección IP destino 192.168.1.1 y dirección IP origen 192.168.1.2.

Los individuos de la población admiten la posibilidad de generalizar alguno de sus genes, asignándose al mismo el valor -1 . Luego el cromosoma redefinido queda:

Ejemplo 2. (0,0,2,5,2114,80,192,168,1,-1,192.168,1,-1)

Este individuo puede encontrar coincidencia con las instancias de tráfico dirigidas u originadas en los nodos de la red comprendidos entre las direcciones IP 192.168.1.0 y 192.168.255.255.

La población inicial es generada a partir de instancias de tráfico del conjunto de entrenamiento, seleccionadas aleatoriamente en base a una función de probabilidad uniforme.

3.2. Función de optimización.

Se propone una función de optimización definida según la ecuación (1):

$$f(r) = \frac{\prod_{j=1}^{m} \prod_{i=1}^{n} \alpha(r_i, d_{ji})}{|D|} \tag{1}$$

Para calcular el valor de fitness del individuo r, se comparan los genes del individuo r con los correspondientes d_{ji} de cada una de la instancias de tráfico perteneciente al conjunto de entrenamiento D mediante la función α definida como:

$$\alpha(r_j, d_{ji}) = \begin{array}{c} w_i \text{ si } r_j = d_{ji} \\ 0 \text{ si } r_j \neq d_{ji} \end{array} \tag{2}$$

Donde cada gen tiene un peso asociado w_i con el objeto de favorecer a aquellos atributos que por experiencia disciplinar resultan más relevantes. En caso de no presentar coincidencias en algún gen, la función α asigna cero. Consecuentemente como resultado de la productoria, la función de fitness penaliza a los individuos que no coincidan en alguno de sus genes asignándoles también un valor cero.

Se introduce una variante α' de la función de peso, definida en la ecuación (3), la cual en lugar de asignar cero cuando algún gen de un individuo no presente coincidencia, le asigna un peso w'_i, que se ajusta en la práctica al valor 0.1

$$\alpha'(r_j, d_{ji}) = \begin{array}{c} w_i \text{ si } r_j = d_{ji} \\ w'_i \text{ si } r_j \neq d_{ji} \end{array} \quad (3)$$

De esta manera la función de fitness, mediante α' favorece a los genes que presentan una frecuencia de aparición relativamente alta. Estos individuos en futuras generaciones, pueden originar nuevas reglas que coincidan con un significativo número de instancias de tráfico.

3.3. Crowding determinístico.

En trabajos anteriores si bien se menciona la utilización de técnicas para garantizar la convergencia hacia múltiples soluciones, fundamentalmente crowding, no se detalla con suficiente profundidad las posibilidades de aplicación al dominio.

En este trabajo se utiliza la técnica conocida como crowding determinístico propuesta originalmente por Mahfoud [7] que introduce la competencia entre padres e hijos por la misma área del espacio de soluciones posibles. Esta elección se basa en que crowding determinístico presenta buenos resultados sin aumentar la complejidad computacional del algoritmo como se menciona en Sereni et. al [9].

Las técnicas como sharing no resultan viables en este caso, ya que el proceso de búsqueda de los individuos de la población que se encuentran en el mismo nicho, resulta demasiado costoso en términos de tiempo computacional. A lo que se agrega la necesidad de recalcular el fitness de todos los individuos en cada generación, lo que eleva en varios ordenes de magnitud el tiempo total de ejecución.

La implementación de crowding determinístico utilizada en este trabajo es descripta por Leon et. al. [10]. En donde una vez realizada la operación de cruzamiento, cada hijo reemplaza a alguno de sus padres más próximo, si y sólo si, aquel posee un valor de fitness más alto.

Como se mencionó en los antecedentes la noción de proximidad está dada por una función de distancia. Mahfoud [7] observa que dicha función debe poseer la mayor cantidad de información posible sobre el dominio de aplicación del problema. Debido a esto se propone la utilización de una función de distancia euclidiana definida en la ecuación (4), que al igual que la función de optimización favorece a través de pesos a ciertos atributos seleccionados en base a la experiencia disciplinaria.

$$\sqrt{(p_i - p_j)^2 w^1 + (s_i - s_j)^2 w^2 + (s'_i - s'_j)^2 w^3 + (d_i - d_j)^2 w^4 + (d'_i - d'_j)^2 w^5} \quad (4)$$

donde p signfica el puerto origen, mientras que s, s' y d, d' son los 2 bytes más significativos de las direcciones IP de origen y destino respectivamente. Los w^k indican los pesos asignados a los términos de la ecuación (4)

3.4. Obtención de los mejores individuos.

Debido a la aplicación de técnicas de nicho, existen dentro de la población grupos o clústeres de individuos que presentan soluciones muy similares. Con el objeto de obtener los mejores representantes de cada uno de estos grupos, se utiliza una heurística cuyo pseudocódigo es presentando en el algoritmo 1.

Algoritmo 1 Heurística para la selección de mejores individuos

```
P=Población
R=conjunto de reglas
N=Número máximo de reglas
for p ∈ P hasta N do
  if p ≠ algún r ∈ R then
    R = R ∪ p
  else
    R = R∪ IndividuoMasGeneralizado(r,p)
  end if
end for
```

El individuo p con mayor fitness se extrae de la población. Se verifica que no exista en el conjunto de reglas algún individuo r con genes de idéntico valor a p. Si ambos individuos presentan coincidencias en sus genes, se prefiere a aquel que observe el mayor número de genes generalizados y se procede a descartar al otro. Este procedimiento se repite hasta que el conjunto de reglas R este integrado por un número N de individuos.

4. Caso de estudio: Aplicación al conjunto de instancias de tráfico de DARPA.

El objetivo de los experimentos es determinar la capacidad del algoritmo propuesto para obtener una población de individuos que presenten coincidencias con el mayor número de instancias de tráfico. Además de comprobar la influencia de la función de fitness utilizada y las funciones de distancias utilizadas en la técnica de crowding.

Para realizar los experimentos se utiliza el conjunto de datos provisto por DARPA[11] el cual ha sido muy utilizado en trabajos dentro del área [2,3,4,10,12,13].

Se ejecutan 40 procesos a lo largo de 1200 generaciones, sobre 2 conjuntos conformados con instancias de tráfico tomadas de los datos provistos por DARPA. El primer conjunto se utiliza para la etapa de entrenamiento y contiene 9000 entradas que representan 4 horas de tráfico. El segundo conjunto se emplea para la fase de prueba y contiene 35000 instancias de tráfico que representan 24 horas.

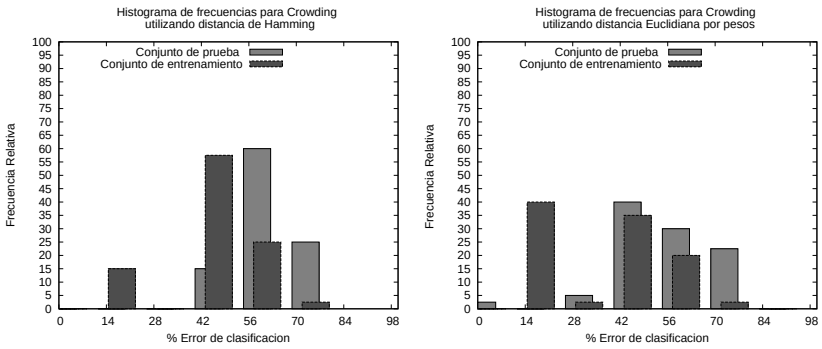
Al finalizar el entrenamiento, los individuos obtenidos son utilizados para buscar coincidencias en el conjunto de prueba y se presenta el porcentaje de errores cometidos. Se considera que un individuo presenta coincidencias con una

instancia de tráfico si y sólo si los atributos no generalizados del mismo coinciden con los atributos correspondientes de la instancia de tráfico.

4.1. Estudio de la influencia de la función distancia en crowding.

En esta subsección se estudia la influencia de las funciones distancia mencionadas en las sección 3.2, para los operadores crowding y crowding determinístico. Los resultados, para los conjuntos de prueba y entrenamiento, se presetan mediante histogramas de frecuencias relativas en función del error porcentual.

Una comparación de los resultados obtenidos utilizando la función distancia Hamming y la función distancia euclideana por pesos, para el operador de crowding clásico, se muestra en la figura 1.



(a) (b)

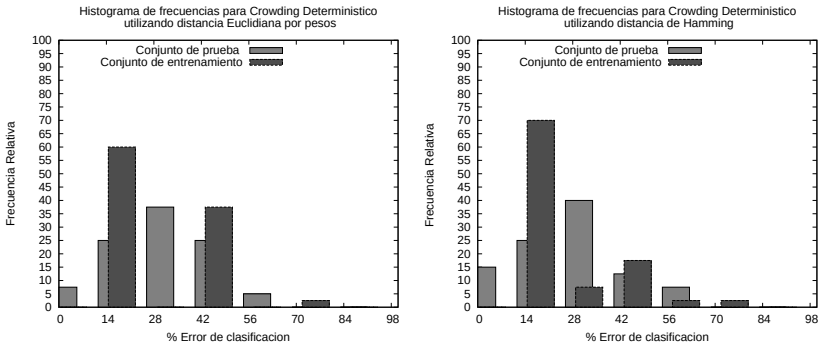
Figura 1. Histograma de frecuencias relativas con los porcentajes de error de clasificación obtenidos variando la función distancia sobre crowding

La figura 1(a) muestra los resultados al aplicar la función distancia de Hamming. Se observa que los valores más altos del histograma se encuentran en el 56 % del error del clasificación para el conjunto de entrenamiento y en el orden del 70 % para el conjunto de prueba. En la figura 1(b) se presentan los resultados al utilizar la función de distancia euclidiana con pesos. Se observa que el valor más alto en el histograma para el conjunto de entrenamiento ahora ha descendido al 28 % de error de clasificación y al 56 % para el conjunto de prueba.

La función de distancia euclidiana basada en pesos ofrece una considerable mejora en el funcionamiento del algoritmo frente a la función de distancia de Hamming. Sin embargo los resultados en el conjunto de prueba resultan desalentadores ya que en muy pocos casos se han obtenidos errores de clasificación inferiores al 50 %.

Los resultados aplicados a crowding determinístico se muestran en la figura 2. En la misma se observa una considerable mejoría respecto a los anteriores. En la figura 2(a) puede verse que alrededor del 60 % de los 40 procesos finalizaron

con un error de clasificación cercano al 15 %. Mientras que para el conjunto de prueba el porcentaje de error obtuvo un 25 %.



(a) (b)

Figura 2. Histograma de frecuencias relativas con los porcentajes de error de clasificación obtenidos variando la función distancia sobre crowding determinístico

Los resultados al aplicar distancia de Hamming se presentan en la figura 2(b). En este caso los porcentajes de frecuencia relativa resultan ser muy similares a los de la figura 2(a) tanto para el conjunto de entrenamiento como para el conjunto de prueba. Luego se desprende que la utilización de la propuesta de función de distancia euclidiana basada en pesos no ofrece ventajas significativas sobre la función de distancia de Hamming al utilizar crowding determinístico.

4.2. Estudio de la influencia de la función de peso en la función de optimización.

En esta subsección se comparan las distintas variantes para la función de peso α y α' desarrolladas en las sección 3.2.

Las reglas obtenidas por el algoritmo genético al aplicar la función de peso α en la función de optimización se muestran en el cuadro 2. Las tres primeras celdas indican el número de regla y las instancias de tráfico que fueron exitosamente reconocidas en el conjunto de entrenamiento (IRE) y conjunto de prueba (IRP). Se observa en este caso que el algoritmo ha convergido a una única solución. Esto se debe a que una gran cantidad de individuos no han sido capaces de encontrar coincidencias en por lo menos una instancia de tráfico del conjunto de entrenamiento y en consecuencia fueron penalizados y eventualmente, descartados.

Los individuos obtenidos al finalizar la ejecución del algoritmo utilizando la función de peso α' se muestran en el cuadro 3, que en este caso conforman 16 reglas. Se observa que muchas de las reglas no han sido capaces de encontrar coincidencias en ninguna instancia de tráfico del conjunto de entrenamiento y

Cuadro 2. reglas obtenidas utilizando α

	IRP	IRE	HH	MM	SS	Protocolo	sPort	sPort	sIP	sIP	sIP	sIP	dIP	dIP	dIP	dIP
1	7620	4120	0	0	1	5	-1	80	172	16	-1	-1	207	-1	-1	-1

el conjunto de prueba. Este resultado responde por un lado a las características de la función de peso α' que ha favorecido a individuos con atributos con alta frecuencia de aparición y por otro a la heurística detallada en la sección 3.4 que selecciona a los individuos que presentan mayores diferencias entre si. Esto ha permitido encontrar reglas que podrían ser capaces de clasificar instancias de tráfico no presentes en el conjunto de entrenamiento.

Cuadro 3. reglas obtenidas utilizando α'

	IRP	IRE	HH	MM	SS	Proto	sPort	dPort	sIP	sIP	sIP	sIP	sIP	sIP	sIP	sIP
1	7184	2912	0	0	1	5	-1	80	172	16	117	-1	-1	-1	-1	-1
2	5086	2669	0	0	1	5	-1	80	172	16	116	-1	-1	-1	-1	-1
3	5592	2658	0	0	1	5	-1	80	172	16	115	-1	-1	-1	-1	-1
4	5004	0	0	0	1	5	-1	80	172	16	112	-1	-1	-1	-1	-1
5	3491	0	0	0	1	5	-1	80	172	16	113	-1	-1	-1	-1	-1
6	0	0	0	0	1	5	-1	80	172	16	60	-1	-1	-1	-1	-1
7	0	0	0	0	1	5	-1	80	172	16	118	-1	-1	-1	-1	-1
8	0	0	0	0	1	5	-1	80	172	16	101	-1	-1	-1	-1	-1
9	0	0	0	0	1	1	-1	80	172	16	116	-1	-1	-1	-1	-1
10	0	0	0	0	1	1	-1	80	172	16	115	-1	-1	-1	-1	-1
11	0	0	0	0	1	7	-1	80	172	16	117	-1	-1	-1	-1	-1
12	0	0	0	0	1	5	-1	84	172	16	117	-1	-1	-1	-1	-1
13	0	0	0	0	1	5	-1	16	172	16	117	-1	-1	-1	-1	-1
14	0	0	0	0	1	4	-1	80	172	16	116	-1	-1	-1	-1	-1
15	0	0	0	0	1	7	-1	80	172	16	116	-1	-1	-1	-1	-1
16	0	0	0	0	1	5	-1	88	172	16	116	-1	-1	-1	-1	-1

Una comparación de los mejores resultados obtenidos con el algoritmo propuesto utilizando las funciones α y α', respectivamente se presentan en la figura 3

Como se observa en la figura 3(a) el algoritmo que utiliza la función α converge a una solución a partir de la generación 800 y obtiene resultados apenas inferiores al 60 % de error en el conjunto de entrenamiento y muy cercanos al 80 % en el conjunto de prueba. La figura 3(b) muestra los mejores resultados obtenidos con el algoritmo cuando se emplea la función α' definida en la ecuación 3. En este caso la convergencia del algoritmo se observa a partir de la generación 800 con una solución cercana al 15 % de error. Mientras que en el conjunto de prueba recién a partir de la generación 900 se converge hacia una solución cercana al 20 % de error.

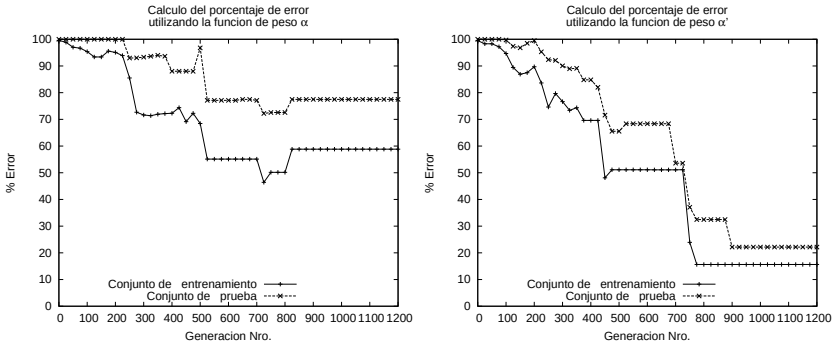


Figura 3. Porcentaje de error en los conjuntos de prueba y entrenamiento utilizando la función α definida en la ecuación (2) y la función α' definida en la ecuación (3)

5. Conclusiones y trabajos futuros.

La adaptación propuesta de un algoritmo genético para el reconocimiento de patrones de tráfico observa resultados prometedores en el caso de estudio planteado, se generan reglas con atributos que permiten encontrar coincidencias del orden del 85 % para el conjunto de entrenamiento y alrededor de un 80 % para el conjunto de prueba.

De las técnicas de crowding evaluadas, crowding determinístico ha presentado los resultados más prometedores a costa de no aumentar la complejidad computacional del algoritmo. Crowding determinístico muestra no ser sensible a la función de distancia utilizada, situación que no se presenta en otras variantes de crowding.

La utilización de la función de peso α' definida en la ecuación (3) constituye una mejora significativa a la función de fitness. Como se ha señalado en la sección 4.2 se han podido encontrar reglas de clasificación que, potencialmente son capaces de clasificar instancias de trafico no presentes en el conjunto de entrenamiento.

Se observa que las funciones de fitness descriptas en el trabajo son costosa en términos de recursos computacionales, por lo que lo resulta de interés estudiar la aplicación de algoritmos genéticos distribuidos [16,17]. Estos algoritmos permiten acelerar la convergencia de las soluciones a la vez que exploran distintos espacios de soluciones.

En trabajos futuros se comprobará el funcionamiento de las reglas obtenidas en la detección de anomalías.

Referencias

1. Mukherjee B., Heberline L. T., Levitt K. *Network Instruction Detection*. IEEE Network, 1994

2. Gong R.H., Zulkernine M., Abolmaesmumi P. *A Software Implementation of a Genetic Algorithm Based Approach to Network Intrusion Detection.* Sixth SNPD/SAWN, IEEE, 2005.
3. Li W. *A Genetic Algorithm Approach to Network Intrusion Detection.* SANS Institute, http://www.giac.org/practical/GSEC/Wei_Li_GSEC.pdf (accedido Mayo 2007)
4. Lu W., Traore I. *Detecting New Forms of Network Intrusion Using Genetic Programming.* Computational Intelligence, vol. 20, pp. 3, Blackwell Publishing, Malden, pp. 475-494, 2004.
5. Goldberg D. *Genetic Algorithms in search Optimization & Machine Learning.* Addison Wesley, 1989
6. Miller B., Shaw M. *Genetic algorithms with dynamic niche sharing for multimodal-function optimization.* IlliGAL report Nro. 95010, University of Illinois at Urbana-Champaign, 1995.
7. Mahfoud, S. *Crowding and Preselection Revisited.* IlliGAL report Nro. 92004, University of Illinois at Urbana-Champaign, 1992.
8. Rao Vemuri V., Cedeno W. *Multi-Niche Crowding for Multi-Modal Search.* Practical Handbook of Genetic Algorithms - New Frontiers, Volume II, CRC Press, 1995
9. Sareni B., Krahenbuhl L. *Fitness Sharing and Niching Methods Revisited.* IEEE Transactions on Evolutionary computation Vol 2. No. 3 September 1998
10. Leon E., Nasraoui O., Gomez J. *Anomaly Detection Based on Unsupervised Niche Clustering with Application to Network Intrusion Detection.* Congress on Evolutionary Computation, 2004. CEC2004. *IEEE*
11. MIT Lincoln Laboratory, DARPA datasets, MIT, USA, http://www.ll.mit.edu/IST/ideval/data/data_index.html (accedido Noviembre 2006).
12. Taeshik S., Yongdue K., Cheolwon L., Jongsub M. *A Machine Learning Framework for Network Anomaly Deteccion using SVM and GA.* Proceedings of the 2005 IEEE Workshop on Information Assurance, 2005
13. Mahoney M. *A Machine Learning Approach to Detecting Attacks by Identifying Anomalies in Network Traffic.* PhD Dissertation, Florida Institute of Techonology, 2003.
14. Bridges S., Vaughn R. *Fuzzy Data Mining and Genetic Algorithms Applied to Intrusion Deteccion.*, National Information Systems Security Conference, 2000
15. Gomez J., Dasgupta D., Gonzalez F., Nasraoui O. *Complete Expression Trees for Evolving Fuzzy Classifier Systems with Genetic Algorithms and Application to Network Intrusion Detection.* Fuzzy Information Processing Society, Proceedings. NAFIPS. 2002 Annual Meeting of the North American. 2002.
16. Alba E., Troya J. *A Survey of Parallel Distributed Genetic Algorithms.* Complexity Volume 4 , Issue 4 , 1999
17. Cantú Paz E. *A Summary of Research on Parallel Genetic Algorithms.* IlliGAL Report No. 95007, University of Illinois at Urbana-Champaign, 1995
18. Sinclair C., Lyn P., Matzer S. *An Application of Machine Learning to Network Intrusion Detection.* 15th Annual Computer Security Applications Conference 1999.
19. Wong, M. L., and K. S. Leung. *Data mining using grammar based genetic programming and applications.* Kluwer Academic Publishers, 2000
20. Crosbie M., and Spafford. G. *Applying Genetic Programming to Intrusion Detection.* AAAI Fall Symposium on Genetic Programming, Nov. 10-12, Cambridge, Massachusetts, 1995.