

# Software de bloqueo automático de páginas web que incitan a la violencia a través de un algoritmo híbrido de aprendizaje computacional\*

## Auto lock software web pages incite violence through a hybrid computational learning algorithm

Jorge Enrique Rodríguez\*\*

Ángela Paola Herrera\*\*\*

Martha Liliana Rojas\*\*\*\*

Fecha de recepción: febrero de 2014

Fecha de aprobación: abril de 2014

### Resumen

El siguiente artículo se basa en un proyecto de investigación que consiste en la creación de un algoritmo híbrido de aprendizaje computacional supervisado para el bloqueo automático de páginas web que incitan a la violencia a través de contenido *snuff*. En este se hace una descripción de la situación actual del problema a resolver y se da a conocer el proceso de selección de los algoritmos para la construcción del algoritmo híbrido y con ello la explicación de su funcionamiento y modelo matemático.

**Palabras clave:** sistema de bloqueo, aprendizaje computacional, violencia en Internet.

---

\* Artículo de investigación.

\*\* Ingeniería en Telemática, Universidad Distrital Francisco José de Caldas, Bogotá D.C, Colombia. Contacto: jrodri@udistrital.edu.co

\*\*\* Ingeniería en Telemática, Universidad Distrital Francisco José de Caldas, Bogotá D.C, Colombia. Contacto: apherred@udistrital.edu.co

\*\*\*\* Ingeniería en Telemática, Universidad Distrital Francisco José de Caldas, Bogotá D.C, Colombia. Contacto: mlrojas@udistrital.edu.co

## Abstract

In this paper we development a hybrid algorithm of machine learning for the automatic locking of web pages that incite to violence through content snuff. We describe of the current situation of the problem to be solved, and we give to know the selection process of the techniques for the development of the hybrid algorithm. Also, we show the mathematical model and results analysis.

**Keywords:** Locking system, Machine Learning, Internet Violence

## Introducción

En países como Colombia, las leyes que rigen la producción y comercialización de material audiovisual resultan insuficientes [1], razón por la cual, las personas pueden acceder fácilmente a contenido violento, sobre todo a través de Internet donde se comparten recursos sin restricción alguna.

Aunque en el presente, las empresas disponen de políticas para el buen uso de los recursos corporativos por parte de los empleados, tener fácil acceso a páginas con contenido violento puede empeorar el consumo del ancho de banda existente y a su vez expone la seguridad de la red ante *malware*. Por tal razón, para un administrador es de suma importancia mantener su red protegida de virus, espías y sobretodo limitar los recursos al uso corporativo objeto sobre el cual fue diseñada. Por ello, en el siguiente artículo se hará una revisión al desarrollo de un sistema de bloqueo automático como herramienta de apoyo al proceso de gestión en una red LAN, puesto que por medio del análisis de un patrón de comportamiento con algoritmos de aprendizaje computacional supervisado, se pueden detectar automáticamente y de manera constante nuevas páginas que, según su experiencia, indiquen contenido *snuff*.

En este artículo se describe el problema, se presenta la explicación del modelo matemático utilizado en el algoritmo híbrido y el resultado de las pruebas realizadas.

## 1. Problema

Internet es una excelente herramienta de comunicación, donde se puede adquirir información de diversas fuentes y con ello multiplicar el conocimiento, facilitando de esta manera que se creen nuevas tendencias hacia grupos que favorecen la investigación y el aprendizaje. Las fuentes generalmente se encuentran en diferentes ubicaciones geográficas, lo que es inadvertido para los cibernautas; pero a su vez, Internet puede ser una tecnología peligrosa, ya que acceder fácilmente a cualquier fuente de información puede conllevar a visitar páginas web de contenido que promocionan diferentes tendencias, por ejemplo el racismo, la xenofobia, terrorismo cibernético, *grooming*, *ciberbullying*, *sexting*, pornografía infantil y trata de personas, entre otros. Además, en alguno de estos sitios web se hacen invitaciones al suicidio y al asesinato a través del sadismo y *snuff* – por ejemplo, grabaciones de asesinatos reales –.

Actualmente, en Colombia las empresas proveedoras de telecomunicaciones, ofrecen productos que bloquean contenido violento,

por ejemplo ETB (Empresas de Telecomunicaciones Bogotá) ofrece el guardián de contenidos, donde los padres pueden administrar las páginas que podrán ser accesibles por sus hijos. Sin embargo, cada día se crean nuevas páginas web de contenido violento, lo que implica ingresar cada una de estas URL manualmente, tarea que resulta ser tediosa.

Igualmente, en las redes LAN los administradores deben hacer una verificación diaria de la creación de páginas Web de contenido violento e ingresarlas manualmente en el *squid* para que estas sean bloqueadas a través de servidores *proxy*; al realizar esta labor algunas veces se carece de tiempo suficiente, lo que favorece la omisión de algunos sitios Web cuyo contenido debería ser restringido.

## 2. Algoritmo híbrido SVM - RBF

El algoritmo SVM es un clasificador lineal y no lineal, el cual ha estado tomando importancia porque se basa en el concepto de la minimización del riesgo estructural<sup>1</sup> y posee una sólida fundamentación matemática en la clasificación y regresión de datos. En este proyecto se aplica el SVM no lineal, ya que en la fase de integración y recopilación se construyó un conjunto de datos con ocho (8) atributos, lo que nos indica que los datos no pueden ser linealmente separables puesto que la información no es fácilmente predecible para separarla en dos clases. A diferencia del SVM lineal, el no lineal adiciona un núcleo o función kernel, el cual aplica una transformación a los datos para que puedan ser divididos, esta función kernel puede convertirse en una debilidad para el algoritmo si la función escogida no es efectiva en el proceso de transformación, por esta razón para

este proyecto se optó por el Kernel gaussiano, es decir, un algoritmo basado en Función Base Radial (RBF).

A continuación se describe el concepto y modelo matemático del SVM lineal y del SVM no lineal con el RBF.

### a. SVM lineal [1]

Las SVM lineal con margen máximo es el modelo más sencillo de SVM y el que tiene menos condiciones de aplicabilidad, debido a que parte del supuesto de que el conjunto de entrada es linealmente separable en el espacio de entrada, es decir, que los ejemplos pueden ser separados por un hiperplano de forma que, en cada lado del mismo, solo queden ejemplos de una misma clase sin hacer ninguna transformación de los datos, como se puede observar en la figura 1 [2].

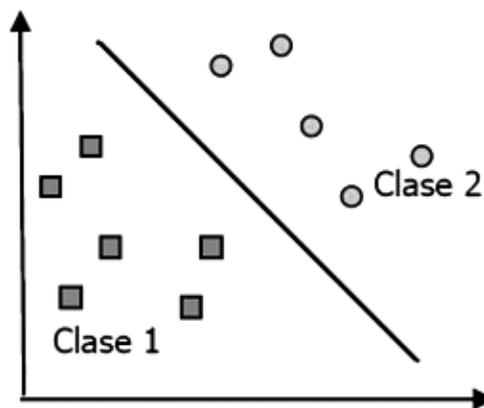


Figura 1. Caso linealmente separable

Fuente: tomado de <http://www.repositorium.org/repositorium/archives.php?id=all&page=26593&sort=author>

Matemáticamente, dadas  $N$  muestras  $x_i$ , con sus correspondientes etiquetas asociadas,  $y_i$ , que definirán a qué clase pertenece cada muestra, tenemos:

<sup>1</sup> Minimización de riesgo estructural (SRM). Minimiza la probabilidad de clasificación errónea sobre nuevos ejemplos, especialmente cuando se disponen de pocos datos de entrenamiento.

$$(x_1 \circ y_1), (x_2 \circ y_2), \dots, (x_N \circ y_N); x_i \in R^J, y_i \in \{+1, =1\} \tag{1}$$

Siendo  $J$  el número de dimensiones o componentes de los vectores que contienen los datos.

Un clasificador es lineal si su función de decisión puede expresarse mediante una función lineal en  $x$ . Así pues, la ecuación del hiperplano de separación será el lugar de los puntos  $x$  en los que se cumple:

$$H : x \circ w + b = 0 \tag{2}$$

Siendo  $b$  una constante que indica la posición del plano respecto al origen de coordenadas. Esta constante recibe el nombre de *sesgo*. es el vector normal al hiperplano y tiene la forma:

$$w = \sum_{i=1}^N \alpha_i y_i x_i \tag{3}$$

Donde los  $\alpha_i$  son los multiplicadores de Lagrange,<sup>2</sup> los cuales como se ha mencionado anteriormente, serán nulos en su mayoría, salvándose únicamente los que sean vectores soporte.

La clasificación se realiza determinando en qué zona del hiperplano está el punto a clasificar. Así pues, el clasificador puede representarse mediante las expresiones:

$$x_i \circ w + b > +1 \text{ para } y_i = +1 \tag{4}$$

$$v. \{ x. \circ w + b \} > 1$$

Si el resultado de la operación es positivo, la muestra pertenecerá a una clase, y si es negativo a la otra.

Las expresiones anteriores pueden combinarse en una única:

$$y_i \{ x_i \circ w + b \} \geq 1 \tag{5}$$

En un problema linealmente separable habrá infinitos hiperplanos que cumplan esta condición. El que buscamos es aquel que tenga un mayor margen. Es decir, queremos maximizar la distancia entre los datos y la frontera de decisión.

Los puntos que caen sobre cada uno de los hiperplanos son los que cumplen:

$$\begin{aligned} H_1 : x_i \circ w + b &= +1 \\ H_2 : x_i \circ w + b &= -1 \end{aligned} \tag{6}$$

Por tanto, estos dos hiperplanos son paralelos entre sí y paralelos al hiperplano  $H$  (2). El margen será la distancia entre  $H_1$  y  $H_2$ . Lo que se busca aquí es aumentar la generalización, es decir, que una vez entrenada la máquina, clasifique correctamente las nuevas muestras. Por ello, cuanto mayor sea la distancia entre los datos y la frontera de clasificación, mejor.

Si se desea que los datos más cercanos a la frontera tengan una salida  $\pm 1$ , la distancia de los datos al plano será  $d = \frac{1}{\|w\|}$ , que será la distancia de  $y$  a  $H$ .

Así pues, el margen del hiperplano  $H$  (distancia a los vectores más cercanos pertenecientes a diferentes clases) a maximizar es  $d = \frac{2}{\|w\|}$ . Maximizar este margen es equivalente a minimizar la norma de  $w$ . De este modo, el problema de encontrar el hiperplano óptimo puede formularse como:

$$L(w) = \min \left\{ \frac{1}{2} \|w\|^2 \right\} \tag{7}$$

<sup>2</sup> Multiplicadores de Lagrange. Es un método para trabajar con funciones de varias variables que nos interesa maximizar o minimizar, y está sujeta a ciertas restricciones.

Con la restricción (5) que garantiza que el hiperplano separará las muestras de distintas clases. Si usamos los multiplicadores de Lagrange e incorporamos la restricción (4), la expresión a minimizar queda de la forma:

$$L_d = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i (y_i (x_i \cdot w + b) - 1) \quad (8)$$

Así pues, para minimizar la expresión anterior debemos derivar con respecto a  $w$  y  $b$  e igualar a cero. Esto nos deja dos ecuaciones:

$$w = \sum_{i=1}^N \alpha_i y_i x_i \quad (9)$$

$$\sum_{i=1}^N \alpha_i y_i = 0$$

Teniendo en cuenta las ecuaciones anteriores, la máquina de vectores soporte, actuando como clasificador y para el caso en el que el problema sea linealmente separable, puede escribirse como:

$$f(x) = \text{sign} \left( \sum_{i=1}^N \alpha_i y_i (x_i \cdot x) + b \right) \quad (10)$$

### Caso no separable linealmente [1]

En el caso que no sea posible separar linealmente los datos, hay dos soluciones: buscar una frontera no lineal, o tratar de encontrar el hiperplano que cometa un menor número de errores.

Para esta última opción, vamos a introducir unas variables positivas  $\xi_i$ , que controlan el error permitido y penalizan las muestras mal clasificadas. Así pues, la ecuación (5) es modificada de la forma:

$$y_i \{x_i \cdot w + b\} \geq 1 - \xi_i \quad (11)$$

$$\xi_i \geq 0, \forall i \quad (12)$$

De este modo, en las muestras bien clasificadas se cumplirá que  $0 < \xi_i < 1$ , dependiendo de qué tan cerca esté de la frontera la muestra, mientras que en las mal clasificadas  $\xi_i > 1$ .

En esta ocasión, la ecuación a minimizar (7), es modificada de la forma:

$$L_p = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \quad (13)$$

Sujeta a las restricciones (11) y (12).  $C$  es una constante que penaliza las muestras mal clasificadas y que determina la importancia de la maximización de la distancia frente a la minimización de los errores. En el presente proyecto, el valor de esta constante se ha calculado mediante validación cruzada, tal y como se explicará en el siguiente capítulo. Usando multiplicadores de Lagrange, la ecuación (13) queda de la forma:

$$L_{pd} = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \mu_i \xi_i - \sum_{i=1}^N \alpha_i (y_i (x_i \cdot w + b) - 1 + \xi_i) \quad (14)$$

Donde es el multiplicador de Lagrange introducido para forzar la positividad de  $\xi_i$ . Si ahora derivamos con respecto a cada variable e igualamos a cero para obtener la solución óptima, como hicimos en el caso separable, obtenemos las ecuaciones (9) y:

$$C - \mu_i - \alpha_i = 0 \quad (15)$$

El problema cuadrático queda de la forma:

$$\dots \quad (16)$$

### SVM no lineal [1]

La SVM no lineal se puede interpretar como una generalización del hiperplano óptimo de decisión, ya que permite la resolución de problemas no separables y trazar fronte-

ras de clasificación no lineales. Para el caso en el que los datos no son linealmente separables, puede aplicarse una transformación  $\phi(x)$  sobre el espacio de trabajo con el objetivo de obtener un espacio de características, generalmente de dimensión superior, donde sí sean separables las muestras y donde se debe trazar el hiperplano óptimo de separación (ver figura 2). La formulación es básicamente la misma, únicamente reemplazando  $x$  por  $\phi(x)$ .

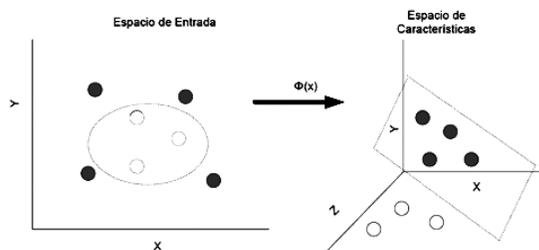


Figura 2. Transformación espacial del espacio de entrada

Fuente: tomado de <http://e-archivo.uc3m.es/handle/10016/9912>

El Teorema de Cover proporciona la justificación de por qué una transformación no lineal a un espacio de mayor dimensión aumenta las posibilidades de disponer de un conjunto de datos separables, si estos no lo eran en el espacio de entrada. Para construir una SVM en el espacio resultante, este debe ser un *Espacio de Hilbert*<sup>5</sup>, y por tanto cumplir:

$$k(x, x_i) = \langle (x), \phi(x_i) \rangle \tag{17}$$

Donde la función  $k$  es llamada *Función núcleo* o *Función kernel*. Teniendo esta función, es posible aplicar el algoritmo de entrenamiento de las SVM sin conocer  $\phi$ . Existen distintas funciones kernel que permiten adaptar la SVM a cada conjunto de muestras, con el fin de obtener mejores resultados. Las más usadas son:

Lineal

$$k(x, x_i) = x \cdot x_i \tag{18}$$

Polinómica

$$k(x, x_i) = (yx \cdot x_i + c)^\alpha \tag{19}$$

Gaussiana

$$k(x, x_i) = \exp\left(-y|x - x_i|^2\right) \tag{20}$$

Siendo una constante de proporcionalidad,  $c$  un coeficiente y  $\alpha$  el rango del polinomio. En el presente trabajo se ha optado por un kernel gaussiano (RBF).

La función de clasificación de la SVM (10) con el RBF es redefinida de la siguiente manera:

$$f(x) = \text{sign}\left(\sum_{i=1}^N \alpha_i y_i (k(x, x_i) + b)\right) \tag{21}$$

El Teorema de Mercer expone que para garantizar que la función sea un producto interno en un espacio de alta dimensionalidad, se debe cumplir la condición:

$$\iint k(u, v) g(v) dudv > 0 \tag{22}$$

Es decir, la función kernel debe ser una función definida positiva. Por tanto, el problema cuadrático de entrenar la SVM queda de la forma:

$$L_d = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j k(x_i, x_j) \tag{23}$$

Sujeto a las restricciones:

$$\begin{aligned} 0 \leq \alpha_i \leq c \\ \sum_i \alpha_i y_i = 0 \end{aligned} \tag{24}$$

El vector del hiperplano resultante es:

$$w = \sum_{i=1}^N \alpha_i y_i \phi(x_i) \tag{25}$$

Se puede dar el caso de que el problema tampoco sea separable en el nuevo espacio. En

estos casos, se actúa de igual manera que en el problema lineal: introduciendo un término de coste  $C$  y las pérdidas  $\xi_i$ . De esta forma, reescribiendo las ecuaciones (11) y (12) para el caso no lineal, obtenemos:

$$y_i \{ \phi(x_i \cdot w + b) \} \geq 1 - \xi_i \tag{26}$$

$$\xi_i \geq 0, \forall_i$$

De este modo, la nueva función objetivo será la misma que en el caso lineal para problemas no separables:

$$\min_{w,b,s_i} \frac{1}{2} \|w\|^2 - C \sum_{i=1}^N \xi_i \tag{27}$$

Si introducimos las ecuaciones (26) en (27), el langrangiano del problema queda de la forma:

$$L_{pd} = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \mu_i \xi_i - \sum_{i=1}^N \alpha_i (y_i (\phi(x_i) \cdot w + b) - 1 + \xi_i) \tag{28}$$

Ecuación similar a (14) en la que únicamente se ha sustituido  $x_i$  por  $\phi(x_i)$ .

Las condiciones que debe cumplir un vector para ser solución de (26) vienen determinadas por el Teorema de Karush-Kuhn-Tucker (KKT) que es una generalización del Teorema de los Multiplicadores de Lagrange. Se puede decir que resolver el problema de la SVM es equivalente a encontrar una solución para las condiciones KKT.

### Pruebas y resultados

Para la realización de las pruebas se dividió el dataset construido en 10 subconjuntos, donde cada uno contenía 2753 registros. A continuación se presenta el resultado comparativo sobre la eficiencia que se obtiene entrenando los datos con el algoritmo SVM, el algoritmo RBF y el híbrido SVM - RBF.

### SVM no Lineal con Kernel Lineal

Se aplica a la base de datos SVM no lineal el Kernel Lineal. Para el Kernel Lineal se requiere sólo un parámetro,  $C$ , que represente la cota superior del vector  $\alpha$ . Se varía este parámetro usando los siguientes 21 valores:

$$C = \{0.1; 0.3; 0.5; 0.7; 1; 10; 20; 30; 40; 50; 60; 70; 80; 90; 100; 200; 300; 400; 500; 700; 1000\}$$

El mejor resultado se obtiene para un coeficiente  $C=100$ , con un promedio de un 94,55% de efectividad en la validación cruzada. En la tabla 1 se muestra el detalle de los resultados obtenidos.

Tabla 1. Validación SVM Kernel Lineal

Validación SVM Kernel Lineal	
Subconjunto	Acierto (%)
1	89,47
2	92,98
3	94,74
4	92,98
5	94,74
6	94,74
7	94,74
8	96,49
9	98,25
10	96,43
Promedio	94,55

Fuente: elaboración propia.

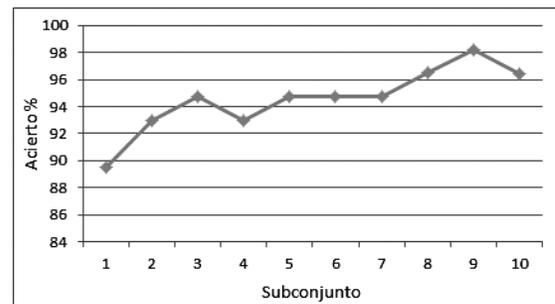


Figura 3. Gráfica SVM Kernel Lineal

Fuente: elaboración propia.

Con el resultado de la gráfica (figura 3) se puede observar que la eficiencia va incrementando constantemente, de tal forma que nos hace deducir que, en cada entrenamiento, el algoritmo SVM aumenta la efectividad de su aprendizaje en cada subconjunto. En el subconjunto 9 se presenta la eficiencia más alta ya que se utilizó el coeficiente  $C = 100$ , el cual es un parámetro ajustable; un valor grande de  $C$  equivale a una mayor penalización de los errores.

### Función Base Radial (RBF)

Se hicieron pruebas colocando desde 2 hasta 20 neuronas para la capa de neuronas cuya función de activación es del tipo radial. Los grupos dependen de la inicialización de los que se realice al principio del algoritmo. Por esta razón, y con el objetivo de encontrar la mejor red posible, se decidió realizar un total de 250 iteraciones para cada uno de los 10 subconjuntos, guardando el resultado de aquella para la cual se obtenga un error de clasificación inferior. En la tabla 2 se muestran los resultados:

Tabla 2. Validación RBF

Validación SVM Kernel Polinomial	
Subconjunto	Acierto (%)
1	71,232
2	72,567
3	72,567
4	75
5	72,567
6	72,567
7	72,567
8	72,982
9	74,206
10	71,214
Promedio	65,254

Fuente: elaboración propia.

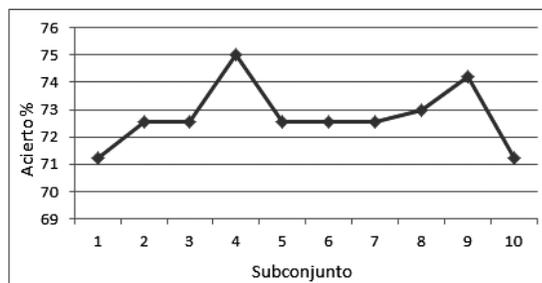


Figura 4. Gráfica RBF

Fuente: elaboración propia

En la tabla 2 se evidencia que la efectividad del RBF en la clasificación de datos es menor que la del SVM. La figura 4 expone que el valor más alto se presentó con en el subconjunto 4, este valor puede darse por los datos que registra el dataset y las neuronas de la capa oculta.

### Hibrido SVM - RBF

El Kernel Gaussiano o RBF, viene dado por la siguiente expresión:

$$k(x_i, x_s) = \exp\left[-\frac{\|x_i - x_s\|^2}{2p^2}\right], p > 0 \quad (29)$$

El parámetro a variar es  $\rho$  y de acuerdo a lo realizado en el trabajo, es decir, se utilizan los valores  $\{0,1, 0,5, 1, 2, 3, 4, 10, 20,100\}$ . La mejor configuración  $\{\rho, C\}$  se tiene para  $\rho = 2$   $C=100$ , con un promedio de un 98.25 % de efectividad en la validación cruzada. En la tabla 3 se muestra el detalle de los resultados obtenidos en la validación cruzada.

Tabla 3. Validación Hibrido SVM - RBF

Validación SVM Kernel RBF	
Partición	Acierto (%)
1	96,491
2	96,491
3	94,737
4	100
5	96,491
6	100
7	100

Validación SVM Kernel RBF	
Partición	Acierto (%)
8	98,246
9	100
10	100
Promedio	98,25

Fuente: elaboración propia.

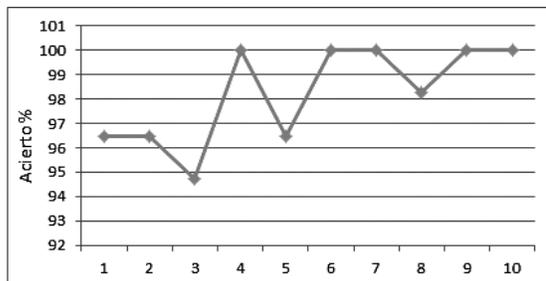


Figura 5. Gráfica Híbrido SVM - RBF

Fuente: elaboración propia.

La gráfica (figura 5) que generó el sistema híbrido muestra que, a medida que transcurre el proceso de aprendizaje del algoritmo, este va aumentando su eficiencia en la clasificación de datos, en cinco subconjuntos se observa un acierto del 100 por ciento, esto podría significar que el parámetro C configurable del SVM no tiene tanta influencia mientras el parámetro p se mantenga en valores bajos.

## Conclusiones

Se realizó una recolección de datos (dataset) sobre páginas web cuyo contenido sea violento, determinando los atributos con base a la descripción principal de la estructura SEO en sitios web. A este conjunto de datos se le aplicó un proceso de preparación, limpieza e integración donde se concluye que sus instancias se encuentran correctamente definidas para ser usadas en el aprendizaje del algoritmo híbrido.

El estudio de fortalezas y debilidades de los cinco algoritmos preseleccionados, permitió determinar que el algoritmo SVM, a pesar de que es una de los más recientes, ha tomado fuerza en la Inteligencia Artificial, ya que su flexibilidad de uso de datos lineales y no lineales lo ha convertido en uno de los algoritmos de clasificación más usados y exactos. En las pruebas realizadas se detectó que en el algoritmo híbrido SVM-RBF se consigue un porcentaje menor de error en la clasificación de datos. En este híbrido se puede ajustar el coeficiente C del SVM y el parámetro p del RBF, jugar con el valor de estos datos ha permitido lograr un acierto del 100 por ciento que no fue posible conseguir con cada uno de los algoritmos por separado.

## Trabajos futuros

Estudiar y comparar otras técnicas de representación y clasificación, con el fin de identificar y converger en una solución que permita en determinado momento implementar este tipo de técnicas en clasificación de otro tipo de contenido Web, de esta forma podrán controlarse todos aquellos contenidos que expongan la seguridad en las redes y por tanto el uso inapropiado de los recursos corporativos hacia los cuales éstas están destinadas.

## Referencias

- [1] F. Valle Padilla, *Implementación eficiente de clasificadores Prior-SVM para MATLAB*. Acceso: enero del 2014. Disponible en <http://e-archivo.uc3m.es/handle/10016/9912>
- [2] M. Márquez Vera, *Clasificación de sedimentos Clásticos mediante Máquinas de Vectores de Soporte*. Acceso: enero del 2014. Disponible en <http://www.repositorium.org/repositorium/archi->

- ves.php?id=all&page=26593&sort=autor>
- [3] *Aplicación de las técnicas de redes neuronales para el diagnóstico on-line del proceso de electroerosión por hilo*. Acceso: septiembre del 2012. Disponible en [http://www.disa.bi.ehu.es/spanish/profesores-etsi\\_bilbo/~jtpcaaxi/PFC/wwwANN/aprndizaje\\_de\\_las\\_ann.htm#\\_Toc136065875](http://www.disa.bi.ehu.es/spanish/profesores-etsi_bilbo/~jtpcaaxi/PFC/wwwANN/aprndizaje_de_las_ann.htm#_Toc136065875)>
- [4] E. Caicedo Bravo, *Funciones de Base Radial*, Universidad del valle Acceso: abril del 2013. Disponible en [http://eisc.univalle.edu.co/materias/Fundamentos\\_De\\_Computacion\\_Inteligente/material/RN\\_RBF\\_Feb2005.pdf](http://eisc.univalle.edu.co/materias/Fundamentos_De_Computacion_Inteligente/material/RN_RBF_Feb2005.pdf)>
- [5] *Técnicas de Análisis de Datos en WEKA*. Acceso: enero del 2014. Disponible en <http://isa.umh.es/asignaturas/crss/tutorialWEKA.pdf><
- [6] O. Comelli, A. Garmendia, *Redes de Funciones de Base Radial*, Instituto Tecnológico Autónomo de México, 2002. Acceso: abril del 2013. Disponible en <<http://cursos.itam.mx/Fakuri/2002/S22002/RNS/Presentaciones/RBF/Funciones/de/base/radial.doc>>
- [7] *LibSVM - A Library for Support Vector Machines*. Acceso: enero del 2014. Disponible en <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>>
- [8] S. Maldonado, R. Weber, "Modelos de Selección de Atributos Para Support Vector Machines", *Revista Ingeniería de Sistemas*, Vol. XXVI, septiembre 2012. Acceso: julio del 2013. Disponible en <http://www.dii.uchile.cl/~ris/RISXXVI/maldonado1.pdf>>
- [9] F. Valle Padilla, *Implementación Eficiente de Clasificadores Prior - SVM para MATLAB*, Universidad Carlos III de Madrid. [Acceso: enero del 2014. Disponible en <http://e-archivo.uc3m.es/handle/10016/9912>>