PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

# Prototipo de un Sistema híbrido IDS/Cortafuego, utilizando Máquinas de Soporte Vectorial

**Daniel Alejandro Araya Astudillo**

TESIS DE GRADO
MAGÍSTER EN INGENIERÍA INFORMÁTICA

Julio 2010

Pontificia Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática

# Prototipo de un Sistema híbrido IDS/Cortafuego, utilizando Máquinas de Soporte Vectorial

**Daniel Alejandro Araya Astudillo**

Profesor Guía: **Broderick Crawford Labrin**

Programa: **Magíster en Ingeniería Informática**

Julio 2010

# ÍNDICE DE CONTENIDOS

## Índice de Figuras

# I. INTRODUCCIÓN

El computador más seguro es el que se encuentra desconectado de la red y que no cuenta con dispositivos de entrada o salida de datos. Es claro que esto es una utopía ya que las necesidades actuales requieren que estos se encuentren conectados con otros equipos para intercambiar información ya sea dentro de una red local o con otras redes. Por otro lado, en la actualidad la información almacenada en estos computadores se ha convertido en un tema relevante para el desempeño de la empresa, debiendo estas realizar grandes inversiones tanto en hardware como en software con la finalidad de asegurar la integridad, disponibilidad y confidencialidad de los datos almacenados [1].

Un computador al encontrarse obligado a estar conectado en una red, implicará que corre el riesgo de sufrir un ataque a su seguridad, ataque realizado por lo que se denomina un intruso y con el fin de robar la información almacenada en dicho computador, modificarla o simplemente impedir que este computador preste los servicios para los cuales fue diseñado.

En la actualidad los ataques que puede sufrir un sistema evolucionan dinámicamente, por lo cual utilizar un Cortafuego basado en reglas estáticas se tornará algo inmanejable por la gran cantidad de reglas que deberá contener o por las constantes actualizaciones que el administrador de dicho sistema tendrá que realizar y aun así sin poder reaccionar con la debida diligencia para evitar que un ataque se complete. Por este motivo en esta tesis se propone un sistema que actualice automáticamente las reglas de un Cortafuego, según las decisiones que tome un Sistema de Detección de Intrusos (IDS) basado en Detección de Anomalías. Para llevar a cabo la diferenciación y clasificación del tráfico normal con el posible ataque se utilizarán las Máquinas de Soporte Vectorial (SVM).

Las Máquinas de Soporte Vectorial (SVM) son una familia de algoritmos desarrollados por Vapnik, que han sido utilizados tanto para problemas de clasificación como de regresión [5]. Estos algoritmos constituyen unas potentes herramientas para el aprendizaje de la relación existente entre la entrada y la salida de un sistema [5]. Debido a la gran eficacia y potencia de las Máquinas de Soporte Vectorial se pueden encontrar un gran número de aplicaciones donde han sido empleadas con gran éxito como son: la minería de datos, la clasificación automática de textos, clasificación de imágenes, reconocimientos de voz, bioingeniería, biometría, y comunicaciones [5].

Algunas de las principales características de las Máquinas de Soporte Vectorial son:

- Las Máquinas de Soporte Vectorial son no paramétricos: Los parámetros del modelos no son predefinidos y su número depende del conjunto de datos de entrenamiento disponible [5]
- La tarea de una Máquinas de Soporte Vectorial consiste en encontrar un hiperplano (frontera de decisión) de N-dimensiones que separe en forma óptima los datos en dos categorías [5]

## II. OBJETIVO GENERAL

Diseñar un Prototipo de Cortafuego cuyas reglas sean actualizadas automáticamente según las decisiones que tome el sistema de detección de intrusos basado en Máquinas de Soporte Vectorial.

## III. OBJETIVOS ESPECÍFICOS

- Comprender el fundamento teórico de las Máquinas de Soporte Vectorial.
- Presentar trabajos relacionados con la implantación de distintos Sistemas de Detección de Intrusos, y utilizando Máquinas De Soporte Vectorial para la clasificación de la data.
- Diseñar un prototipo de Cortafuego con generación de reglas en forma automática al detectar un tráfico anómalo.
- Realizar pruebas de análisis con un prototipo de servidor.
- Analizar los resultados obtenidos.

## IV. HIPÓTESIS

Detectar y prohibir oportunamente el análisis de puertos de servicios evitará que un atacante identifique tanto las aplicaciones y el sistema operativo utilizado en un servidor, por lo cual no podrá detectar las posibles vulnerabilidades que posea dicho sistema y con ello evitando que explote estas.

Implantar un Prototipo de Cortafuego cuyas reglas sean definidas en base a las decisiones que toma un sistema de detección de intrusos basado en anomalías y utilizando máquinas de soporte vectorial, permitirá detectar en forma oportuna los posibles ataques a un servidor, mejorando con ello la seguridad de este, evitando con ello la pérdida de información sensible.

## V. METODOLOGÍA DE TRABAJO

La presente tesis se ha desarrollado a través de un trabajo principalmente práctico que ha involucrado la integración de distintas herramientas de un ambiente shell en Linux, tales como: Libsvm, Iptables, TCPDUMP, etc. El prototipo se implementó en un ambiente Shell de Linux, debido a que es uno de los principales sistemas operativos utilizados en los servidores de Internet. La razón de utilizar Libsvm, se basa principalmente que es una potente herramienta compatible con la programación Shell muy utilizada en distintos estudios.

Por otro lado, el prototipo utiliza las Máquinas de Soporte Vectorial para la toma de decisión y clasificación del tráfico en uno anormal o normal, y basándose en dicha clasificación creará la regla respectiva en el cortafuego que prohibirá todo tráfico desde un computador. Para entrenar las Máquinas de Soporte Vectorial se generaron consultas HTTP desde un cliente al servidor. Una vez entrenadas las Máquinas de Soporte Vectorial se procede a comprobar el sistema en conjunto para lo cual se genera tráfico desde un cliente determinado hacia el Servidor y se observa si se ha tomado la decisión correcta creando o no la regla en el cortafuego.

Cada vector fue construido con 22 atributos que corresponden a los campos que contiene tanto el protocolo IP, adicionalmente al encabezamiento TCP y UDP. Posteriormente, a través de la utilización de herramientas como WEKA o Excel, se observa que algunos atributos se podrán podar, sin embargo esto no se realiza en el entrenamiento de las Máquinas de Soporte Vectorial, por temas de completitud.

La presente tesis se organiza de la siguiente manera: primero se presentan distintos estudios realizados sobre la materia propuesta. Posteriormente se presentarán los conceptos teóricos nombrados durante el desarrollo del trabajo. En una tercera sección se presenta el diseño, pruebas y resultados obtenidos con la solución propuesta. En una cuarta sección, se entregan conclusiones. Por último, se anexan los valores calculados de TPR y FPR, y los algoritmos programados.

## VI. MOTIVACIÓN

Tradicionalmente, una máquina que preste algún servicio ya sea este Web, Mail, FTP, será implementado con el servicio que presta y adicionalmente un Cortafuego. Cuando recibe alguna solicitud de servicio, Primeramente se analizarán las reglas que posee el Cortafuego, observando si hay alguna que prohíba la conexión. Si no existiese, entonces la consulta accedería a dicho servicio. Dichas reglas generalmente son estáticas, donde el administrador debe constantemente actualizarlas, lo cual es muy difícil de lograr a cabalidad; ya sea, por la dinámica que siguen los distintos atacantes, por el tiempo de reacción que conlleva al detectar un ataque o por falta de conocimiento por parte del administrador.

Una forma de configurar este Cortafuego, es simplemente prohibiendo el acceso a cualquier puerto de servicio distinto al que se presta. Lo cual logra un nivel de seguridad mínimo. El siguiente ejemplo graficará los inconvenientes de contar solo con un Cortafuego cuyas reglas sean estáticas:

> El Administrador de un servidor Web, configurará las reglas del Cortafuego permitiendo solo el acceso de las direcciones IP validas en Internet (y la red local) al protocolo y puerto del servicio Web (TCP/80). En caso de un ataque, el administrador deberá analizar los registros del Cortafuego y del servidor; actualizar las reglas prohibiendo el acceso a las direcciones IP de los atacantes. Esta acción es netamente reactiva, con un periodo de tiempo extenso: desde que comienza el ataque, se han hecho sentir los efectos del mismo en cuanto a la perdida del servicio, luego se realizan análisis del registro y si se tiene un administrador con el suficiente conocimiento y experiencia, entonces recién se procederá a actualizar las reglas del Cortafuego, cuando el daño en el servicio prestado se ha hecho sentir.

Lo que se propone en esta tesis, es crear un sistema íntegro de seguridad donde:

- Se detecten intentos de ataques, para ello se implanta un Sistema de Detección de Intrusos cuya clasificación de la data recaerá en las Máquinas de Soporte Vectorial.
- Una vez que se clasifique una data como anómala se proceda a actualizar las reglas el Cortafuego en forma automática, prohibiendo el acceso desde el atacante al servidor en cualquier servicio, en el momento en el que el atacante genere un tráfico distinto a lo indicado como normal. Con lo cual se evita que se llegue a explotar alguna vulnerabilidad existente en el computador.
- Entregar una herramienta al administrador para revertir cualquier decisión tomada por el sistema, y con ello minimizar los falsos positivos resultantes.

## VII. TRABAJOS RELACIONADOS

**1.- Sistema OTAD**

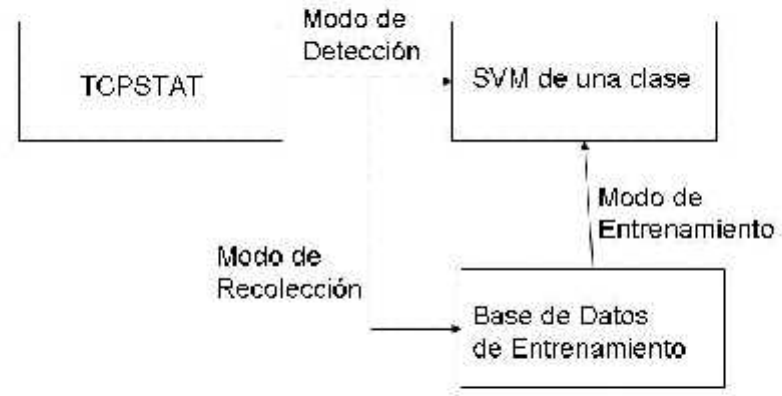


Figura 1: Arquitectura del sistema OTAD

En [25] se presenta un estudio en que se utiliza la herramienta TCPSTAT, para calcular las estadísticas del tráfico de una red en ciertos instantes, y este resultado es enviado para su clasificación a Las Máquinas de Soporte Vectorial.

TCPSTAT es una herramienta que monitorea y reporta cierta estadística sobre una interfaz de red. Esta herramienta puede obtener la información monitoreando directamente una determinada interfaz de red o leyendo dicha información desde un archivo que ha sido almacenado por TCPDUMP.

La conveniencia de utilizar dicha herramienta en este estudio radica en que el resultado del análisis estadístico es formateado como un vector en texto plano, el cual puede ser enviado directamente a las Máquinas de Soporte Vectorial de una clase, sin exigir un excesivo tratamiento previo. TCPSTAT puede calcular hasta veinte tipos de estadística, pero en dicho estudio se utilizaron solamente cinco, estos son: Cantidad de paquetes ICMP, Cantidad de

paquetes UDP, Cantidad de paquetes TCP, Promedio de tamaño de paquetes y la desviación en el tamaño de cada paquete.

La función de Base Radial (RBF) fue utilizada como función Kernel en las Máquinas de Soporte Vectorial de una clase.

Por otro lado, en los experimentos se utilizó la data almacenada por Massachusetts Institute of Technology (MIT), Lincoln Lab, en su evaluación de detección de intrusos DARPA de 1999. Dicha data fue capturada en dos semanas, en las cuales en la primera semana se capturó data con tráfico normal, es decir, libre de ataques y en la segunda semana la data capturada contenía ataques. Por lo cual, en el sistema OTAD se utilizó la data de la primera semana para la fase de entrenamiento de las Máquinas y la data de la segunda semana para la fase de pruebas de dicho sistema.

Los autores de dicho estudio definieron un evento de tráfico de red como un tráfico de red con una duración dada, el cual estaría descrito como el vector de salida de TCPSTAT. Por lo cual el número de eventos estaría dado por la duración de los ataques. Los autores dan el siguiente ejemplo: si la duración es de 60 segundos, entonces el tamaño del conjunto de entrenamiento es de 6601, el tamaño del conjunto de pruebas es de 6424 y el número de eventos de ataques es de 137. En el estudio se presentan los resultados obtenidos para una duración de 60 segundos, 120 segundos, 180 segundos y 300 segundos.

En el estudio se observa que a mayor duración de los eventos, se lograba un mejor funcionamiento del Sistema OTAD. Pero se perdía la opción del análisis de tiempo real del Sistema. Asimismo, para una duración de 300 segundos se lograba un nivel de detección del 71% con un 10% de falsas alarmas.

**2.- Sistema CT-SVM**

En [2] se presenta un estudio donde se propone un método basado en la detección de anomalías y en una solución escalable para redes con este tipo de detección. Se utilizan las Máquinas de Soporte Vectorial para realizar la clasificación.

Los autores de dicho estudio pretenden mejorar el tiempo de entrenamiento cuando se trabaja con grandes volúmenes de datos en las Máquinas de Soporte Vectorial, al dividir el análisis en una jerarquía de grupos. Para el manejo de los grupos (cluster) se utiliza el algoritmo Árbol Autoorganizado de Crecimiento Dinámico (DGSOT). El análisis en grupos ayuda a encontrar los puntos límites, los cuales corresponden a los más representativos para entrenar a las Máquinas de Soporte Vectorial entre dos clases.

En este estudio, se construye un árbol de grupos jerárquico por cada clase en el conjunto de datos utilizando el algoritmo DGSOT. En el algoritmo DGSOT, la agrupación es realizada de arriba hacia abajo, y se construye el árbol jerárquico iterativamente en varias épocas. Después de cada época, los nuevos nodos se añaden al árbol sobre la base de un proceso de aprendizaje. Para evitar la sobrecarga de cálculo de la construcción del árbol, no se construyen los árboles jerárquicos en forma completa. En su lugar, después de cada época se forman las Máquinas de Soporte Vectorial en los nodos de cada árbol, con el fin de controlar el crecimiento de los árboles. En concreto, a los vectores soportados se les permite crecer, mientras que otro tipo de vectores son detenidos. Esto tiene el efecto de la adición de nodos en las zonas de frontera entre las dos clases, mientras se eliminan los nodos distantes de la frontera.

Para realizar las pruebas, en dicho se estudio se utilizó la captura de tráfico realizada por MIT Lincoln Lab, denominada DARPA en 1998. Esta data posee cuatro tipos de ataques más el tráfico normal, por lo cual se tiene un problema de clasificación de 5 clases. Para la implementación de Máquinas de Soporte Vectorial se ocupó la herramienta LIBSVM con una Función de Base Radial como función Kernel, eligiéndose los parámetros de configuración basándose en observaciones, y para el análisis de las cinco clases se procede a un esquema de uno contra uno, debido a la reducción de data necesaria que conlleva en la etapa entrenamiento en comparación a un esquema de uno contra todos.

Según los autores de dicho estudio, los resultado obtenidos demuestran que el sistema CT-SVM es asintóticamente más rápido que utilizar sólo Máquinas de Soporte Vectorial, la tasa de falsos negativos es más baja y la tasa de falsos positivos es igual de buena en ambos sistemas.

**3.- Sistema IPS-SVM**

En [24] se presenta un esquema híbrido, donde primeramente la data de entrada es analizada en un sistema de prevención de intrusos (IPS) basados en reglas, utilizándose para ello la aplicación Snort_Inline.

A fin de minimizar falsas detecciones (falsos positivos) y la pérdida de detección (falso negativo) en la detección de uso indebido, el sistema de prevención de intrusos que se propuso en dicha investigación se compone de Máquinas de Soporte Vectorial con módulos de clases múltiples entrenadas con los resultados de la detección basada en reglas, los módulos se clasifican en 4 clases (verdaderos positivos: identificar el ataque como un ataque, falsos positivos: identificar los valores normales como ataque, verdaderos negativos: identificar los valores normales como normal, falsos negativos: identificar ataque como normal). La estructura del sistema es:

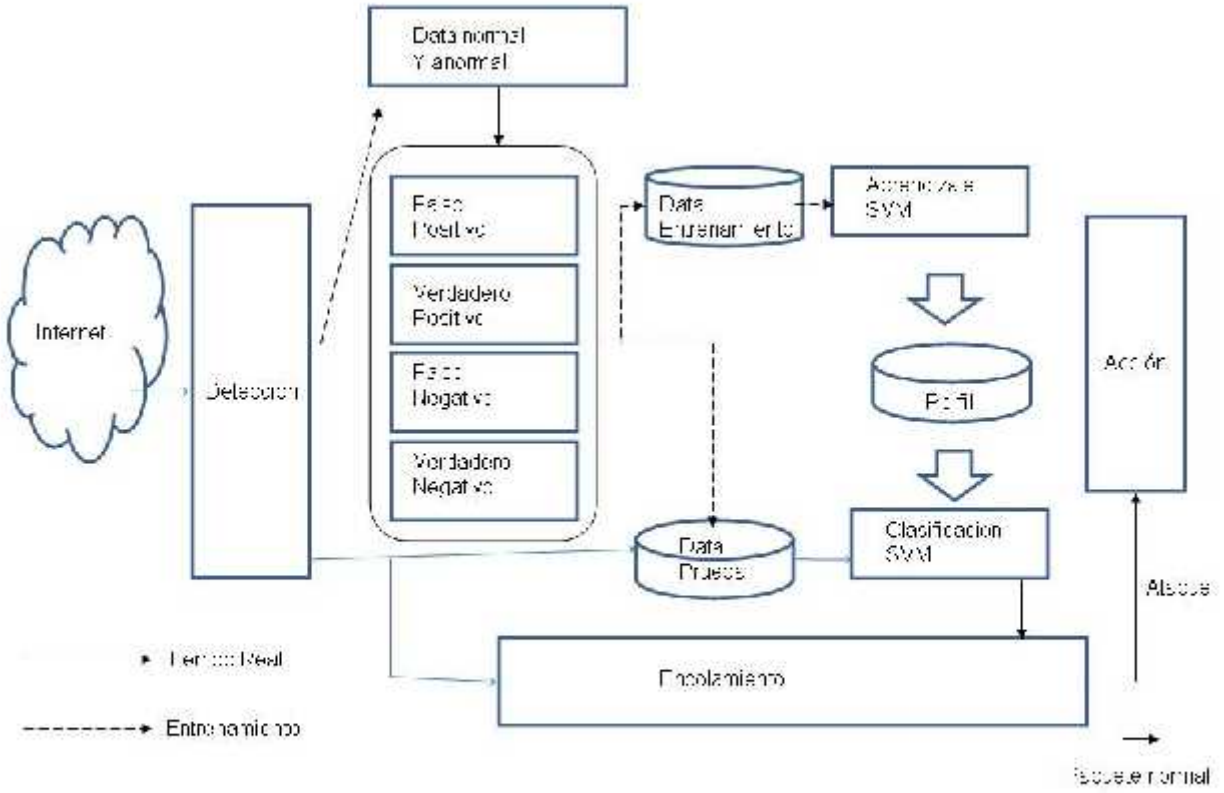


Figura 2: Estructura de sistema de prevención de intrusos basado en Máquinas de Soporte Vectorial

En el sistema de prevención de intrusos, la recolección de paquetes es hecho con Snort_Inline, en modo promiscuo, y la clasificación del tráfico normal o no, es realizada por reglas en la ingeniería de detección.

Debido a que los paquetes detectados incluyen detección de paquetes falsos, así como detección de paquetes perdidos, estos son filtrados a través del módulo Máquinas de Soporte Vectorial para la clasificación multiclase de acuerdo con las 4 clases entrenadas previamente, y sólo los paquetes identificados como verdaderos positivos y falsos negativos son manejadas por la ingeniería respectiva de una manera segura.

La data utilizada durante este estudio fue la captura de tráfico realizada por MIT Lincoln Lab., denominada DARPA en 1998. Un total de 26 características fueron extraídas del encabezamiento de los paquetes. Se crearon dos modelos de Máquinas de Soporte Vectorial de multiclases para minimizar las alarmas falsas. Un modelo utilizó la técnica de

uno contra uno y el segundo modelo utilizó el método uno contra todos. Por otro lado, durante los experimentos se compararon las funciones Kernel, de Base Radial y de Polinomio.

Según los resultados, al utilizar el método de las Máquinas de Soporte Vectorial uno a uno, se lograron mejores resultados que utilizando el método uno contra todos y esto se debe probablemente a las características de las Máquinas de Soporte Vectorial destinadas a la clasificación binaria. Los patrones de clasificación se obtienen con mayor precisión en clases con los datos en igual proporción. Además, al utilizar un parámetro de grado 4 en la función Kernel polinominal, la tasa de clasificación de las cuatro clases fue mayor (84,91%).

Como conclusión de este estudio los autores demuestran que con un método de uno contra uno se logra una mayor precisión.

**4.- Enfoque de múltiples pasos para la evaluación de intrusiones en la red de tráfico.**

En [23] se focaliza un estudio comparativo de distintos esquemas de sistemas basados en detección de anomalías, para detectar algún tipo de intrusión. Varios sistemas supervisados, no supervisados y sus variaciones, tales como, vecino más cercano, el enfoque basado en Mahalanobis, el sistema del factor de valor extremo local (LOF), y Máquinas de Soporte Vectorial sin supervisión.

Estos sistemas son evaluados utilizando un conjunto de datos de DARPA 1998 así como data de redes reales. Dicha se comparación se basó en técnicas estándar de evaluación usando algunas métricas específicas.

La evaluación de los sistemas de detección de intrusos es una tarea difícil debido a varias razones. En primer lugar, es problemático obtener datos de alta calidad para realizar la evaluación debido a la privacidad. En segundo lugar, aunque los datos reales se encuentren disponibles, etiquetar las conexiones de red como normal o intrusivas requiere gran cantidad de tiempo para muchos expertos humanos. En tercer lugar, el cambio constante del tráfico de red no sólo ha introducido nuevos tipos de intrusiones, sino también puede cambiar los aspectos del comportamiento "normal", con lo que la construcción de puntos de referencia útil será aún más difícil. Por último, al medir el rendimiento de un IDS, hay una necesidad de medir no sólo la tasa de detección (por ejemplo, cuántos atentados se detectó correctamente), sino que se requiere saber la tasa de alarmas falsas (cuántas de las conexiones normales fueron erróneamente detectadas como ataques), así como el costo de los errores de clasificación. La evaluación se complica aún más por el hecho de que algunos de los ataques (por ejemplo, la denegación de servicio (DoS), sondaje) pueden usar cientos de paquetes o de conexiones de red, mientras que en los ataques como U2R (usuario root) y R2L (mando a distancia a local) suelen utilizar sólo una o unas pocas conexiones.

Por lo anterior, se han desarrollados algunas métricas estándar para la evaluación de intrusiones en la red, por lo general, estas corresponden a la tasa de detección, así como la tasa de falsas alarmas. La tasa de detección se calcula como el cociente entre el número de ataques detectados correctamente y el número total de ataques, mientras que las tasas de falsas alarmas (falsos positivos) se calculan como el cuociente entre el número de conexiones normales que están clasificadas erróneamente como ataques y el número total de conexiones normales.

En general, hay dos tipos de ataques en la red de detección de intrusos: los ataques que implican conexiones individuales y los ataques que involucran a múltiples conexiones (ráfagas de conexiones). La métrica estándar de tratar todos los tipos de ataques como similares no puede proporcionar una evaluación lo suficientemente genérica y sistemática de los ataques que involucran a muchas conexiones de red (ataques a ráfagas). En particular, no se reúne la información sobre el número de conexiones de red asociado con un ataque que se ha detectado correctamente. Por lo tanto, se deberá aplicar un análisis independiente para cada tipo de ataque según corresponde a un ataque de una conexión o de ráfaga de conexiones. Sin embargo, el primer paso para ambos tipos de análisis corresponde a calcular el valor de la puntuación para cada conexión de red. Donde el valor de la puntuación representa la probabilidad de que la conexión de red en particular esté asociada con una intrusión.

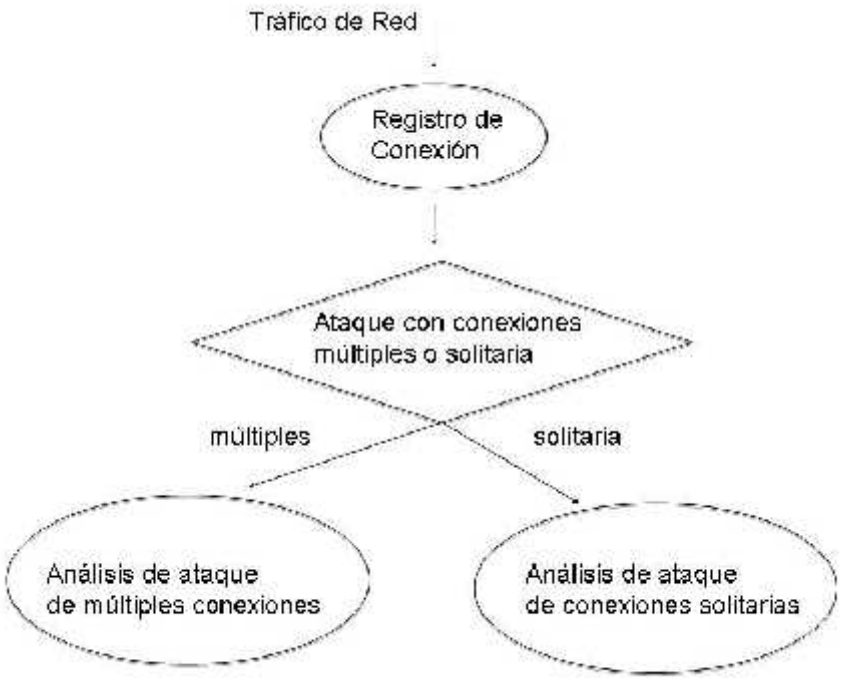


Figura 3: Enfoque de múltiples pasos para la evaluación de intrusiones en la red.

En el enfoque de múltiples pasos que se muestra en la Figura 3 se utiliza un error calculado para cada conexión a fin de obtener métricas adicionales de evaluación. La primera métrica derivada corresponde a las áreas de superficie entre la curva de ataque real y la curva de ataque prevista. Cuanto menor sea la superficie entre estas dos curvas de ataque, el algoritmo de detección de la intrusión será mejor. Sin embargo, la superficie en sí misma no es suficiente para captar muchos aspectos relevantes de los algoritmos de detección de intrusos (por ejemplo, cuántas conexiones están asociados con el ataque, la rapidez del algoritmo de detección de intrusos, etc.) Por lo tanto, algunas métricas adicionales se utilizaron para apoyar la métrica básica de la superficie bajo la curva de ataque. Suponga que el número total de conexiones de red en el conjunto de datos considerado es N. El número N es igual a la suma del número total de conexiones de red normal (Nn) y el número total de conexiones de red que están asociados con las intrusiones (Ni). El número (NFA) corresponde al número de las conexiones no intrusivas (normal) que tienen el puntaje más alto que el umbral especificado de antemano. Ahora bien, las métricas adicionales se pueden definir de la siguiente manera:

- Tasa de detección de ráfaga (BDR): se define para cada ráfaga y representa la relación entre el número total de conexiones de red intrusivas Ni que tienen el puntaje más alto que el umbral especificado de antemano en el ataque a ráfagas y el número total de conexiones dentro de los intervalos de ataque.
- El tiempo de respuesta: representa el tiempo transcurrido desde el inicio del ataque hasta el momento de la primera conexión de red.

Se utilizó TCPTRACE para extraer información sobre los paquetes de las conexiones TCP y la construcción de nuevas características. Los datos de entrenamiento DARPA de 1998 incluye información tal como: identificación de las marcas de tiempo (hora de inicio y duración), tipo de servicio, dirección IP de origen y puerto de origen, dirección IP de destino y el puerto de destino, así como el tipo de cada ataque. Se utilizó ésta información para asignar los registros de conexión, y etiquetar correctamente cada registro de conexión como "normal" o un tipo de ataque.

Dado que la mayoría de los ataques DoS y de sondaje pueden utilizar cientos de paquetes o conexiones, en dicho estudio se construyeron algunas características basadas en el tiempo que tratan de captar las conexiones anteriores con características similares. Sin embargo, existen algunos tipos de ataques que pueden tener intervalos de minutos u horas, por lo cual, estos ataques no pueden ser detectados utilizando características derivadas basadas en el tiempo. Con el fin de capturar a estos tipos de los ataques, se utilizan características basadas en conexión que capturan características similares de los registros de conexión en las últimas 100 conexiones de la misma fuente.

Para apoyar la aplicabilidad de sistemas de detección de anomalías, se aplica un procedimiento para extraer el contenido de características basadas en estadística útil y temporal. Los resultados experimentales realizados con el conjunto de datos indican que las técnicas de detección de anomalías de mayor éxito fueron capaces de lograr una tasa de detección del 74% para ataques con múltiples conexiones y una tasa de detección del 56% para los ataques de conexión única, manteniendo la tasa de falsas alarmas en el 2%. Por otro lado, cuando la tasa de falsas alarmas está

cercana al 4%, se logra una tasa de detección del 89% alcanzada por los ataques a ráfagas, y una tasa de detección del 100% en el caso de los ataques de una sola conexión.

En dicho estudio se concluye que la técnica más prometedora para la detección de intrusos con los datos utilizados es el enfoque valor extremo local (LOF). Además, al realizar experimentos con datos reales de la red, el enfoque LOF tuvo mucho éxito en la selección de varios ataques. Se realizaron experimentos que demuestran que, por ejemplo, las Máquinas de Soporte Vectorial sin supervisión fueron muy prometedoras en la detección de intrusiones nuevas lográndose una tasa de detección muy alta pero a su vez la tasa de falsas alarmas también era muy alta.

Los datos generados por el seguimiento del tráfico de red tienden a tener un volumen, dimensionalidad y heterogeneidad muy alto, por lo que el rendimiento de los algoritmos de minería de datos para el análisis en tiempo real es aun inaceptable.

**5.- Modelo híbrido de Árboles de Decisión y Máquinas de Soporte Vectorial**

En [26] se presenta un nuevo IDS híbrido, donde se mezclan Árboles de Decisión y Máquinas de Soporte Vectorial. La motivación para usar el enfoque híbrido es mejorar la precisión del sistema de detección de intrusiones, en comparación al uso de los enfoques individuales. El enfoque híbrido combina los mejores resultados de los diferentes sistemas individuales que resultan en una mayor precisión.

Un sistema híbrido inteligente utiliza el enfoque de integrar el aprendizaje o diferentes modelos de toma de decisiones. Cada modelo de aprendizaje funciona de una manera diferente y explota un conjunto diferente de características. La integración de los diferentes modelos de aprendizaje proporciona un mejor rendimiento que el aprendizaje individual o modelos de decisión mediante la reducción de sus limitaciones individuales y la explotación de sus diferentes mecanismos. En un sistema híbrido inteligente jerárquico cada capa proporciona una nueva información al nivel superior. El funcionamiento global del sistema dependerá de la correcta funcionalidad de todas las capas. La figura 4 muestra la arquitectura del sistema híbrido inteligente con Árboles de Decisión y Máquinas de Soporte Vectorial

El conjunto de datos se pasa primero a través de los Árboles de Decisión y un nodo de información es generado. El nodo de información se determina según las reglas generadas por los Árboles de Decisión. Los nodos terminales se numeran de izquierda a derecha comenzando con el 1. Todos los registros del conjunto de datos se asignan a uno de los nodos terminales, que representan la clase particular o subconjunto. Los nodos terminales representaran la data ya sea normal o ataque. Esta información de nodo (como un atributo adicional), junto con el conjunto original de atributos se pasa a través de las Máquinas de Soporte Vectorial para obtener el resultado final. La idea clave aquí es

investigar si la información proporcionada por el nodo de los Árboles de Decisión mejorará el rendimiento de las Máquinas de Soporte Vectorial.

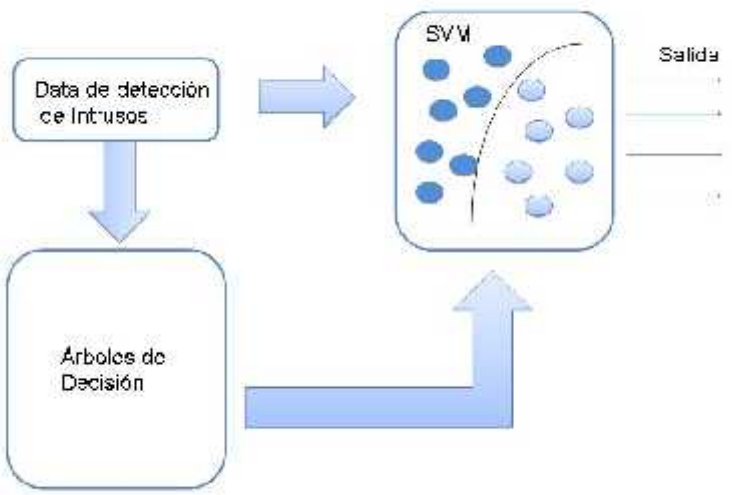


Figura 4: Modelo híbrido de Árboles de Decisión y Máquinas de Soporte Vectorial

Las observaciones empíricas muestran que diferentes tipos de clasificadores proporcionan información complementaria sobre los patrones a clasificar. Aunque para un caso particular, un clasificador Problema funciona mejor que el otro, un conjunto de patrones mal clasificados no necesariamente se superponen. La idea no es contar con un único clasificador de decisión sobre una intrusión; sino que la información de distintos clasificadores individuales se combinan para tomar la decisión final, que es popularmente conocido como el enfoque de conjunto. La eficacia del enfoque de conjunto depende de la precisión y la diversidad de los clasificadores base.

Por lo cual se utilizaron los clasificadores individuales Árboles de Decisión, Máquinas de Soporte Vectorial y un sistema híbrido Árboles de Decisión y Máquinas de Soporte Vectorial. De acuerdo con lo realizado en la etapa de entrenamiento se le asigna a cada clasificador diferentes pesos, según el nivel de resultados obtenidos a las salidas de los clasificadores. Por ejemplo, para la clase 1 si los Árboles de Decisión dan mejores resultados, seguido por el sistema híbrido y por último las Maquinas de Soporte Vectorial, entonces a los Árboles de Decisión se le asignará el mayor peso, seguido por el modelo híbrido y a las Maquinas de Soporte Vectorial se le asignará el peso más bajo. Así para las cinco diferentes clases cada clasificador podrá tener distintos pesos en función de su rendimiento con los datos de entrenamiento. Por lo cual para un registro de datos en particular, si todos los clasificadores tienen opiniones diferentes, sus resultados se consideran y la puntuación más alta se declara como la decisión efectiva del enfoque de conjunto.

En dicho estudio se experimentó con la data utilizada en KDD Cup 1999. Este conjunto de datos cuenta con 41 atributos para cada registro de conexión más una etiqueta de clase. En los ataques R2L y U2R no presentan patrones secuenciales como en un ataque de tipo DOS y de sondaje, debido a que los primeros ataques se ocultan en los paquetes de datos y los ataques de tipo DOS y de sondaje presentan muchas conexiones en un corto período de tiempo. Por lo tanto, algunas características que buscan un comportamiento sospechoso en los paquetes de datos no se construyen y se llaman funciones de contenido. Los experimentos tienen dos fases, la formación y de prueba. En la fase de entrenamiento el sistema construye un modelo utilizando los datos de entrenamiento para conseguir una precisión de máxima generalización (con una precisión en los datos no visto). Los datos de prueba se pasan por el modelo construido para detectar la intrusión en la fase de prueba. Además de los cuatro diferentes tipos de ataques antes mencionados también se debe detectar la clase normal. El conjunto de datos para los experimentos contenían 11.982 registros, que se genera de forma aleatoria a partir de los datos del MIT. Este conjunto de datos se dividen a su vez en datos de entrenamiento con 5092 registros y datos de prueba con 6890 registros. Todos los modelos de detección de intrusos son capacitados y evaluados con el mismo conjunto de datos. Como el conjunto de datos cuenta con cinco clases diferentes que realizar una clasificación de 5 clases. Los datos pertenecen a: clase 1 normal, sondaje pertenece a la clase 2, denegación de servicio (DoS) pertenece a clase 3, usuario root (U2R) pertenece a clase 4 y remoto a local (R2L) pertenece a clase 5. Los experimentos se realizaron con un procesador AMD Athlon, 1,67 GHz procesador con 992MB de RAM.

**Los Árboles de Decisión**

Aunque los Árboles de Decisión son capaces de manejar un problema de clasificación de 5 clases, se utilizó un clasificador de Árboles de Decisión binario, de manera que las comparaciones con el clasificador de la Máquinas de Soporte Vectorial que es un clasificador binario, tendría sentido. Se construyeron cinco diferentes tipos de clasificadores. Los datos se divide en las dos clases de “normal” y “ataque”, donde ataque es la colección de las cuatro clases de ataques (Sondaje, DOS, U2R y R2L). El objetivo es separar normal y los patrones de ataque. Repetimos este proceso para las cinco clases. En primer lugar un clasificador se construyó utilizando los datos de entrenamiento y, a continuación con los datos de las pruebas se probó en el clasificador construido para clasificar los datos en normal o ataque.

**Máquinas de Soporte Vectorial**

La elección de la función Kernel definirá el espacio de características en el que el conjunto de entrenamiento será clasificado. Se utiliza una función Kernel polinomial para los experimentos.

Como las Máquinas de Soporte Vectorial manejan problemas de clasificación binaria de cada clase, se emplean cinco Máquinas de Soporte Vectorial para la detección de los cuatro tipos de ataques y el tráfico normal. El clasificador aprende de los datos de entrenamiento y se utiliza en los datos de prueba para clasificar los datos en los patrones normales o ataques. Este proceso se repite para todas las clases.

Nuestros experimentos muestran que los Árboles de Decisión obtienen una mayor precisión en los ataques de tipo sondaje, R2L y U2R en comparación con las Máquinas de Soporte Vectorial, pero obtiene una menor precisión para la detección de la clase de ataque DoS. Para la clase normal ambos métodos logran el mismo rendimiento. Por los resultados obtenidos se concluye que los Árboles de Decisión logran una buena precisión con pequeños conjuntos de datos de entrenamiento. El tiempo de entrenamiento y los tiempos de las pruebas también son menores para los Árboles de Decisión en comparación con las Máquinas de Soporte Vectorial.

**Modelo Híbrido**

Un modelo híbrido de Árboles de Decisión y Máquinas de Soporte Vectorial (DT-SVM) tiene dos etapas para la construcción del clasificador. Los conjuntos de datos fueron aprobados primero por los Árboles de Decisión y se generó la información del nodo. Los datos de ensayos en conjunto con la información del nodo se entregan a las Máquinas de Soporte Vectorial. Las Máquinas de Soporte Vectorial entregan el resultado final del modelo híbrido.

El rendimiento del modelo híbrido funciona mejor que los Árboles de Decisión y Máquinas de Soporte Vectorial por separado para la clase normal. Para las clases de sondaje, U2R y R2L se obtuvieron mejores resultados que en un enfoque individual de las Máquinas de Soporte Vectorial. De los resultados anteriores se puede concluir que, si bien la información del nodo generado por los Árboles de Decisión hizo aumentar el rendimiento de las Máquinas de Soporte Vectorial, en conjunto, el modelo híbrido no dio el rendimiento esperado

Por otro lado, los resultados empíricos muestran que un enfoque en conjunto logrará un mejor rendimiento para la detección de sondajes y los ataques U2R que los tres modelos individuales.

El enfoque en conjunto clasifica correctamente la mayoría de ellos recogiendo todas las clases, que están clasificados correctamente por todos los tres clasificadores. Como era de esperar el enfoque se basa en las diferencias de grupo de clasificación y mejoran el rendimiento global. Para aprovechar el rendimiento de los diferentes tipos de

clasificadores un sistema inteligente híbrido jerárquico se propone como se muestra en la figura 5. El modelo híbrido de Sistema de Detección de Intrusos hace uso de modelos individuales, modelos híbridos y de enfoques en conjuntos para maximizar la eficiencia computacional y precisión en la detección de cada clase. El modelo propuesto logra la mejor precisión en el funcionamiento general sobre los ataques.

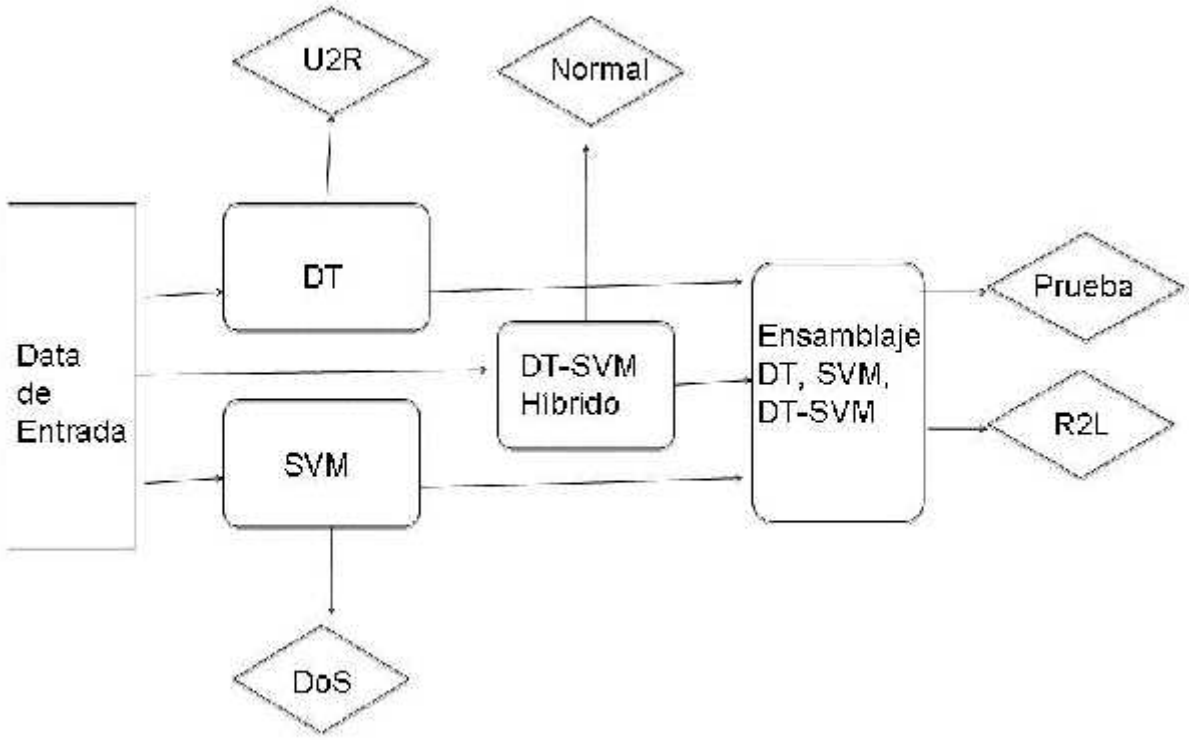


Figura 5: Sistema de Detección de Intrusos basado en un sistema jerárquico inteligente

El enfoque en conjunto logró una precisión del 100% para la clase de sondaje, y esto sugiere que si se eligen adecuadamente los clasificadores se podrían obtener una precisión del 100% en otras categorías también. Por último, se propone un modelo jerárquico inteligente IDS para hacer un uso óptimo de las mejores actuaciones emitidas por los clasificadores individuales y el enfoque en conjunto.

# VIII. CONCEPTOS RELACIONADOS

### 1.- Máquinas De Soporte Vectorial

Las Máquinas de Soporte Vectorial (SVM) son una familia de algoritmos que fueron desarrollados por Vapnik (1995), para solucionar problemas de clasificación, aunque también ha sido extendido su uso a resolver problemas de regresión. Por lo cual, suele utilizarse los términos SVC (Support Vector Clasificación) y SVR (Support Vector Regression) para especificar si se busca una clasificación o una regresión, respectivamente, al utilizar las Máquinas De Soporte Vectorial [5].

En la presente tesis, se utilizarán Máquinas De Soporte Vectorial en la clasificación de la data, es decir, se usará SVC. Por lo cual a continuación se explicará solo el concepto relacionado con SVC.

#### a.- SVC

El problema de clasificación puede limitarse a la consideración del problema de dos clases sin pérdida de generalidad. En este problema el objetivo es separar las dos clases por una función que se induce a partir de ejemplos disponibles. El objetivo es producir un clasificador, que funcionará bien en los ejemplos vistos, es decir, se generaliza bien [20].

Considerando el ejemplo en la Figura 6. Se presentan muchos posibles clasificadores lineales que pueden separar los datos, pero sólo hay uno que maximiza el margen (maximiza la distancia entre éste y el punto más cercano de datos de cada clase). Este clasificador lineal se denomina el hiperplano de separación óptimo [20].

La figura 6, representa un hiperplano que separa un problema de clasificación de dos clases [20]:

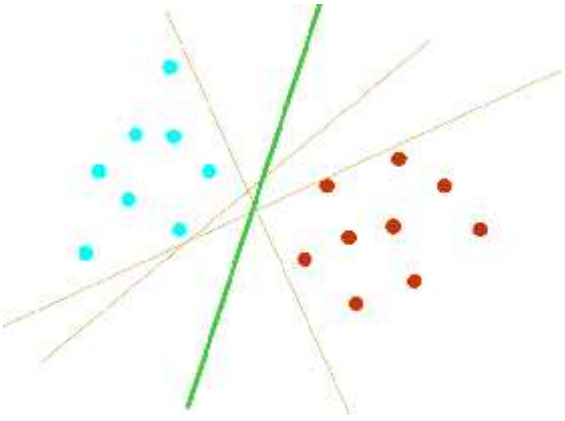

Figura 6: Hiperplano Óptimo Lineal.

Por lo cual, el principio de SVC consiste en construir un hiperplano óptimo lineal que permita separar las clases procurando que [5]:

- Los vectores que pertenecen a las distintas clases se encuentran en distintos lados del hiperplano.
- La mínima distancia (margen) entre los vectores y el hiperplano (frontera de decisión) sea maximizado.

Por otro lado, en términos de Máquinas De Soporte Vectorial una variable predictor es llamado un atributo, y un atributo transformado que usado para definir el hiperplano es conocido como una característica. La tarea de elegir la representación más conveniente se conoce como selección de características. Un conjunto de características que describe un caso (es decir, una fila de los valores del predictor) se llama un vector [5].

**a.1.- El Hiperplano De Separación Óptimo**

Considere el problema de separar el conjunto de vectores de entrenamiento en dos clases,

$$D = \left\{(x^1, y^1),....,(x^l, y^l)\right\}, x^i \in \Re^n, y \in \{-1,1\}, i = 1..l \qquad \text{(Ec 1)}$$

Con un hiperplano,

$$(w \cdot z) + b = 0 \qquad \text{(Ec 2)}$$

Un conjunto de vectores serán separados óptimamente por un hiperplano, si se logran diferenciar en ambas clases sin errores y la distancia entre los vectores más cercanos al hiperplano es la máxima. Sin pérdida de generalidad, la ecuación (2) se puede considerar para un hiperplano canónico. Por lo cual se restringen los parámetros (*w*,*b)* de la siguiente forma [14]:

$$\min_{i=1,..,r} \left| \left(w, x^i\right) + b \right| = 1 \qquad \text{(Ec 3)}$$

En la figura 7 se entrega una representación de los hiperplanos canónicos, indicándose los vectores más cercanos al hiperplano [20].

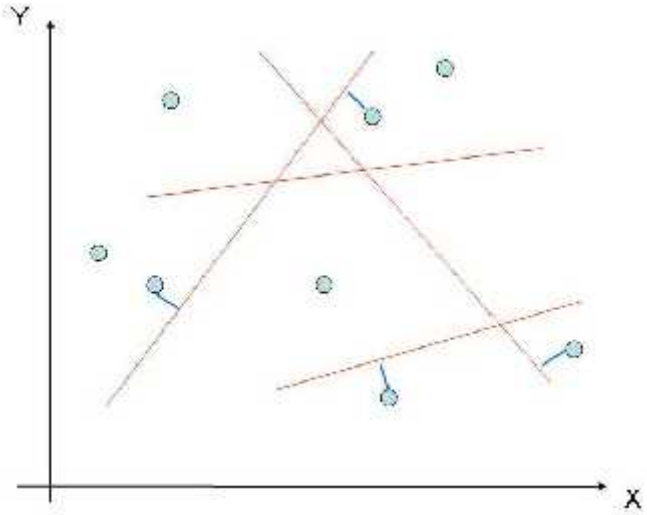


Figura 7: Hiperplano Canónico.

Un hiperplano en forma canónica debería satisfacer las siguientes restricciones:

$$y^i\left(\left(w, x^i\right)+b\right)\geq 1, \quad i=1,..l \tag{Ec 4}$$

La distancia *d(w,b;x)* de un punto *x* al hiperplano *(w,b)* es,

$$d(w,b;x)=\frac{\left|\left(w, x^i\right)+b\right|}{\|w\|} \tag{Ec 5}$$

El hiperplano óptimo se logra maximizando el margen, *p*, sujeto a las restricciones de la ecuación (4), el margen estará dado por:

$$\begin{aligned} p(w,b) &= \min_{x,y=-1} d\left(w,b;x^i\right)+\min_{x,y=1} d\left(w,b;x^i\right) \\ &= \min_{x,y=-1} \frac{\left|\left(w, x^i\right)+b\right|}{\|w\|}+\min_{x,y=1} \frac{\left|\left(w, x^i\right)+b\right|}{\|w\|} \end{aligned}$$

$$= \frac{1}{\|w\|}\left( \min_{x,y=-1} \left| \left(w, x^i\right) + b \right| + \min_{x,y=1} \left| \left(w, x^i\right) + b \right| \right)$$

$$= \frac{2}{\|w\|} \qquad \text{(Ec 6)}$$

Por lo tanto, el hiperplano que separa óptimamente la data, puede ser encontrado reduciendo al mínimo:

$$^2(w) = \frac{1}{2}\|w\|^2 \qquad \text{(Ec 7)}$$

La ecuación 7, es independiente de *b*, debido a que la ecuación 4 siempre se cumple (es decir, es un hiperplano de separación) y los cambios en *b* se moverá en la dirección normal a sí misma. En consecuencia, el margen se mantiene sin cambios, pero el hiperplano ya no corresponderá al óptimo, ya que estará más cerca de una clase que el otro.

Para solucionar este problema de optimización convexo se deberá utilizar Lagrange:

$$L(w, b, \sim) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{r} \sim_i \left( y^i \left( \left( x^i, w \right) + b \right) - 1 \right) \qquad \text{(Ec 8)}$$

Con $\sim_i \geq 0$ , *L* deberá ser maximizado con respecto a $\sim_i$ , y minimizado con respecto a *w* y *b*. Por lo que se tendrá:

$$\frac{¢}{¢w} L(w, b, \sim) = \frac{¢}{¢b} L(w, b, \sim) = 0 \qquad \text{(Ec 9)}$$

Conduciendo a:

$$\sum_{i=1}^{r} \sim_i \cdot y^i = 0 \qquad \text{(Ec 10)}$$

$$w = \sum_{i=1}^{r} \sim_i y^i x^i \qquad \text{(Ec 11)}$$

De las ecuaciones 8, 10 y 11 se obtiene que el problema de optimación convexo esta dado por:

$$\max[\ L(\sim)\ ] = \max_{\sim} -\frac{1}{2}\sum_{i=1}^{r}\sum_{j=1}^{r}\sim_i \sim_j y^i y^j \left(x^i, x^j\right) + \sum_{k=1}^{r}\sim_k \qquad \text{(Ec 12)}$$

Y la solución estará dado por:

$$\sim^* = \arg\min_{\sim} \frac{1}{2}\sum_{i=1}^{r}\sum_{j=1}^{r}\sim_i \sim_j y^i y^j \left(x^i, x^j\right) - \sum_{k=1}^{r}\sim_k \qquad \text{(Ec 13)}$$

Con las siguientes restricciones:

$$\sum_{j=1}^{r}\sim_j y^j = 0 \qquad\qquad \sim_i \geq 0, \quad i = 1,..,r \qquad \text{(Ec 14)}$$

Resolviendo las ecuaciones 13 y 14 se determinarán los multiplicadores de Lagrange, y el Hiperplano de separación óptima estará dado por la ecuación 11 y ecuación 15:

$$b^* = -\frac{1}{2}\left(w, x^r + x^s\right) \qquad \text{(Ec 15)}$$

Donde $x^r$ y $x^s$ son algún vector soportado de cada clase clasificada.

$$\sim_r, \sim_s \rangle 0, \qquad y^r = -1, y^s = 1 \qquad \text{(Ec 16)}$$

Entonces un clasificador estricto se puede encontrar con:

$$f(x) = \text{sgn}\left((w, x) + b\right) \qquad \text{(Ec 17)}$$

Alternativamente, un clasificador no tan estricto puede ser usado interpolando linealmente el margen:

$$f(x) = h\left((w, x) + b\right) \qquad \text{(Ec 18)}$$

Donde,

$$h(z)=\begin{cases} -1 \longrightarrow z<-1 \\ z \longrightarrow -1 \le z \le 1 \\ 1 \longrightarrow z>1 \end{cases} \qquad \text{(Ec 19)}$$

El clasificador dado por las ecuación 18 es más apropiado de utilizar en comparación a un clasificador estrito, cuando se requiere consultar por un clasificador dentro del margen donde no existe una data de entrenamiento.

Cuando se ha encontrado el valor más alto del margen de hiperplano, únicamente los puntos que están más cerca al hiperplano tendrán un $\sim_i \rangle 0,$ y estos puntos se denominan vectores soportados. Todos los otros puntos tendrán un $\sim_i = 0$. Esto significa que sólo aquellos puntos que están más cerca al hiperplano darán la representación del clasificador. Es relevante hacer notar que el margen se verá afectado solo por lo vectores soportados [2].

Por otro lado, si los datos son separables linealmente todos los vectores soportados se encontrarán en el margen y, por tanto el número de vectores soportados puede ser muy pequeño. En consecuencia, el hiperplano es determinado por un pequeño subconjunto del conjunto de entrenamiento, los otros puntos se podrían eliminar de la serie de entrenamiento y volver a calcular el hiperplano produciría la misma respuesta [20].

**a.2.- Generalizando el Hiperplano**

Hasta el momento se ha tratado el caso en que se podía separar linealmente la data. Sin embargo, esto no siempre es posible, en la siguiente figura se presenta un caso donde existen dos atributos que no son posibles de separar:

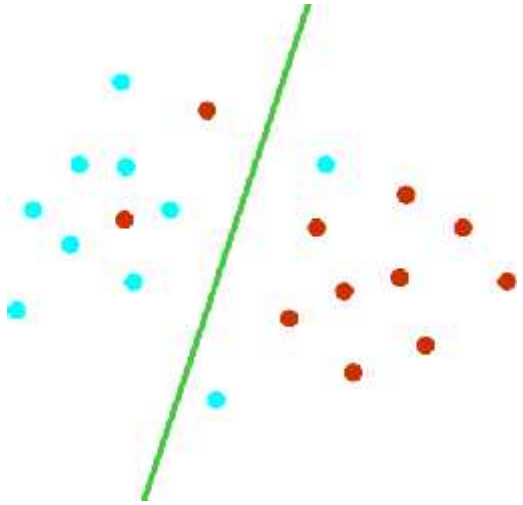

Figura 8: Data no separable linealmente

Existen dos enfoques para generalizar el problema, que dependen del conocimiento previo del problema y del ruido estimado en los datos. Por un lado, en el caso en que se espera (o incluso se sabe) que un hiperplano puede separar los datos correctamente, es adecuado introducir una función de costos adicionales asociados a la clasificación errónea. De otra forma se puede usar una función más compleja para describir la frontera.

Para poder generalizar el Hiperplano de Separación Óptimo Cortés y Vapnik (1995) introdujeron variables de holgura $(\P)$ y es una medida de los errores en la clasificación [14]:

$$\P_i \geq 0, \quad i = 1,..r \qquad \text{(Ec 20)}$$

Y las ecuaciones (13) y (7) se ven modificadas como sigue:

$$y_i \cdot \left(\left(x^i \cdot w\right) + b\right) \geq 1 - \P_i, \quad i = 1,..l \qquad \text{(Ec 21)}$$

$$^2\left(w, \P\right) = \frac{1}{2}\|w\|^2 + ¥ \sum_{i=1}^{r} \P_i \qquad \text{(Ec 22)}$$

El término $\sum_{i=1}^{r} \P_i$ puede ser tomado como algún tipo de medida del error en la clasificación. Mientras que el término $¥$ puede ser definido como un parámetro de regularización (SRM, Structural Risk Minimization), el cual es el único parámetro libre de ser definido en las Máquinas de Soporte Vectorial [11]. El ajuste de este parámetro se puede lograr haciendo un balance entre la maximización del margen y la violación a la clasificación.

Entonces la solución al problema de optimización de la ecuación 22 y bajo las restricciones dadas por la ecuación 21, es obtenida por el punto convexo de Lagrange:

$$^2\left(w, b, \P, \sim, \quad\right) = \frac{1}{2}\|w\|^2 + ¥ \sum_{i=1}^{r} \P_i - \sum_{i=1}^{r} \sim_i \left(y_i \cdot \left(\left(x^i \cdot w\right) + b\right) - 1 + \P_i\right) - \sum_{j=1}^{r} \quad _j \P_j, \qquad \text{(Ec 23)}$$

Donde ~ y son los multiplicadores de Lagrange. Para solucionar el problema se deberá minimizar la ecuación 23 con respecto a *w*, *b*, ¶ y maximizar con respecto a los multiplicadores de Lagrange. La minimización de ² con respecto a *w*, *b*, ¶ se obtienen de:

$$\frac{\partial ²}{\partial b} = 0 \longrightarrow \sum_{i=1}^{r} \sim_i y_i = 0 \qquad \text{(Ec 24)}$$

$$\frac{\partial ²}{\partial w} = 0 \longrightarrow w = \sum_{i=1}^{r} \sim_i y_i x_i \qquad \text{(Ec 25)}$$

$$\frac{\partial ²}{\partial ¶} = 0 \longrightarrow \sim_i + \ _i = 0 \qquad \text{(Ec 26)}$$

El problema dual indicado en las ecuaciones 12 y 13, se aplicarán de igual forma, salvo la siguiente restricción:

$$0 \leq \sim_i \leq ¥ , \qquad i = 0,....., r \qquad \text{(Ec 27)}$$

La solución del problema de minimización se desarrolla de igual forma que un caso linealmente separable excepto por la modificación de los multiplicadores de Lagrange.

**a.3.- Generalizando en un Espacio de Características de una Mayor Dimensión**

En el caso en que un límite lineal sea inapropiado, las Máquinas de Soporte Vectorial pueden mapear el vector de entrada, *x*, en un espacio de características, *z*. Eligiendo, a priori, un mapeo no lineal, las Máquinas de Soporte Vectorial construyen Hiperplano de Separación Óptima en este espacio de alta dimensión.

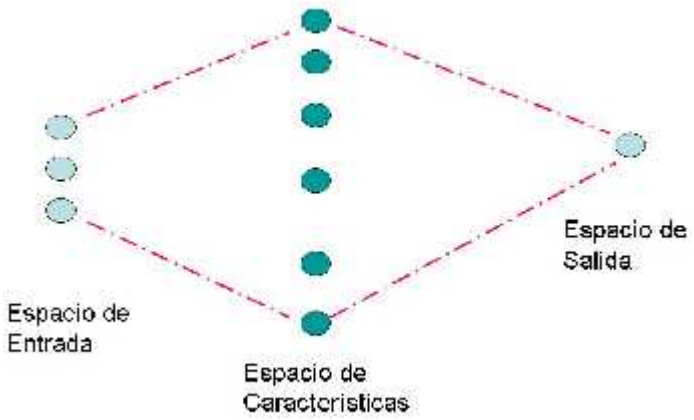


Figura 9: Mapeo de un vector de entrada en un espacio de características [20]

Por lo cual la ecuación 13 se verá modificada de la siguiente manera:

$$\tilde{}^{*} = \arg\min_{\tilde{}} \frac{1}{2}\sum_{i=1}^{r}\sum_{j=1}^{r}\tilde{}_i\tilde{}_j y^i y^j K\left(x^i, x^j\right) - \sum_{k=1}^{r}\tilde{}_k \qquad \text{(Ec 28)}$$

Donde *K* corresponde a la función Kernel que ejecuta el mapeo no lineal al espacio de características, manteniéndose las restricciones de la ecuación 27.

Solucionando la ecuación 27 y 28, el clasificador estricto de la ecuación 17 se verá modificado para implementar el Hiperplano de Separación Óptima en el espacio de características:

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{r}\tilde{}_i y_i K(x_i, x) + b\right) \qquad \text{(Ec 29)}$$

Y las ecuaciones 11 y 15 tendrán las siguientes modificaciones:

$$(w^*, x) = \sum_{i=1}^{r}\tilde{}_i y^i K(x^i, x) \qquad \text{(Ec 30)}$$

$$b^* = -\frac{1}{2}\sum_{i=1}^{r}\tilde{}_i y^i \left[K(x^i, x^r) + K(x^i, x^s)\right] \qquad \text{(Ec 31)}$$

**a.4.- Funciones Kernel**

La idea de las funciones Kernel es habilitar operaciones que sean ejecutadas en el espacio de entradas más que en el potencial espacio de características de alta dimensión. De esta forma el producto interno no necesita ser calculado en el espacio de características. Sin embargo, los cálculos computacionales aún son críticamente dependientes de la cantidad de patrones de entrenamientos y proveer una buena distribución de la data para un problema de alta dimensión requerirá un largo conjunto de entrenamiento [20]. Las principales funciones Kernel son:

i) Polinominal

$$k\left(x, x^{'}\right) = \left[\left(x, x^{'}\right) + 1\right]^{d} \qquad \text{(Ec 32)}$$

ii) Función de Base Radial Gaussiana

$$k\left(x, x^{'}\right) = \exp\left[\frac{-\left\|x - x^{'}\right\|^{2}}{2\pm^{2}}\right] \qquad \text{(Ec 33)}$$

iii) Función de Base Radial Exponencial

$$k\left(x, x^{'}\right) = \exp\left[\frac{-\left\|x - x^{'}\right\|}{2\pm^{2}}\right] \qquad \text{(Ec 34)}$$

iv) Perceptrón de Multicapa

$$k\left(x, x^{'}\right) = \tanh\left(p\left(x, x^{'}\right) + \wp\right) \qquad \text{(Ec 35)}$$

v) Series de Fourier

Pueden ser consideradas una expansión en el siguiente 2N+1 dimensión del espacio de características. El Kernel es definido en el intervalo $\left[-\frac{®}{2},\frac{®}{2}\right]$,

$$k(x,x') = \frac{\sin\left(N+\frac{1}{2}\right)(x-x')}{\sin\left(\frac{1}{2}(x-x')\right)} \quad \text{(Ec 36)}$$

**a.5.- Normalización de la data**

La normalización de la data es requerida en algunas funciones Kernel, debido al dominio restringido en que habitan, Para determinar si la normalización de la data es requerida, se debe analizar las características de la data de entrada.

**a.6.- Máquinas de Soporte Vectorial Multiclase**

Inicialmente las Máquinas de Soporte Vectorial fueron desarrolladas para resolver problemas binarios. Sin embargo existen diferentes estrategias que permiten desarrollar técnicas de Máquinas De Soporte Vectorial para resolver problemas de N-clases. Entre ellas se destacan [18]:

- Uno contra todos: en esta estrategia, se construyen N modelos de Máquinas de Soporte Vectorial. La i-esima Máquina de Soporte Vectorial es entrenada con la muestra de entrenamiento de la clase i-esima con etiqueta positiva y todas las otras muestras de entrenamiento con etiquetas negativas.

- Uno contra uno: este método construye N(N-1)/2 clasificadores donde cada uno es entrenado usando patrones de dos clases. Para clasificar una medida se implementa un sistema de voto. Si la ecuación 29 dice que X pertenece a la clase I, entonces el voto para la clase I se incrementa en uno, si no, el voto para la clase J se incrementa en uno. En caso de igual número de votos entre diferentes clases, se selecciona aquella con índices más pequeños.

- Gráfico acíclico directo (Direct acyclic graph): esta fase es similar al método uno contra uno. En la fase de prueba se usa un gráfico acíclico binario con N(N-1)/2 nodos internos y N hojas. Cada nodo es una Máquinas De Soporte Vectorial binaria de clases i-esima y j-esima. Con una nueva medida X, empezando el nodo en la raíz. Se evalúa la función indicador binario. Entonces se mueve a la derecha o hacia la izquierda dependiendo del resultado. El método procede hasta que se alcanza el nodo de una hoja, que indica la clase predicha.

**a.7 Estrategias de Selección de Variables**

Usando un criterio de selección de variables, la dimensionalidad de los datos puede reducirse sin perder información útil, y al mismo tiempo la información compuesta por ruido puede minimizarse. En definitiva, para estar seguro de que los resultados obtenidos sean buenos, es necesario seleccionar cuidadosamente las variables (parámetros) que se utilizarán junto a los algoritmos de reconocimiento de patrones que se deseen aplicar [5].

Cualquier procedimiento para la selección de las variables basa su funcionamiento en dos aspectos fundamentales: un criterio de selección y procedimiento de búsqueda. Para evitar la explosión exponencial de una búsqueda exhaustiva, se han desarrollado diferentes métodos que exploran el espacio de las variables de una manera más eficaz. Estas estrategias de búsqueda pueden agruparse en tres grandes categorías: exponenciales, secuenciales (o deterministas), y aleatorias (o estocásticas) [5].

- Las técnicas exponenciales realizan una búsqueda cuya complejidad crece exponencialmente con el número de variable. Entre éstos, el método “Branco anda bound” es uno de los más populares. En él se garantiza encontrar el subconjunto óptimo de un tamaño dado, si la función de la evaluación tiene un comportamiento monotónico. En otras palabras, si un clasificador que utiliza un subconjunto de variables de entrada presenta un éxito de clasificación peor que otro clasificador que utiliza otro subconjunto de variables, se asume que ninguna combinación de las variables presentes en el primer subconjunto conducirá a un mejor éxito en la clasificación y, por lo tanto, debe abandonarse la búsqueda entre esas variables [5].

- Los algoritmos de búsqueda secuenciales siguen estrategias que reducen el número de estados que se analizan durante la búsqueda, aplicando la búsqueda local. Los métodos más comunes son [5]:

    - Forward Selection (SFS): Este método comienza con un conjunto sin variables y en forma secuencial va agregando parámetros. El procedimiento continúa hasta que el criterio de selección haya alcanzado un mínimo en el cálculo del error de predicción o todos los parámetros se agreguen al modelo. El procedimiento empieza considerando cada una de las variables individualmente y seleccionando las variables, que da el mejor valor obtenido por el criterio de selección, donde el criterio de selección se calcula por medio del error de predicción sobre los datos de validación.

- Backward Selection (SBS): Este método funciona de forma contraria a SFS, en este caso, todas las variables del conjunto son incluidas al principio para ser utilizadas por el clasificador. Las variables en este caso se van descartando o eliminando en un momento dado basándose en su contribución al criterio de selección. Es decir, se van eliminando secuencialmente aquellas variables cuya exclusión no degrada el cálculo del error de predicción del clasificador.

- Los métodos estocásticos permiten realizar búsquedas locales alrededor de soluciones prometedoras pero poseen la componente de aleatoriedad que les permite explorar otras soluciones en el espacio de búsqueda. El método estocástico más empleado para abordar problemas de optimización es [5]:
  - Algoritmos Genéticos: son procesos de búsqueda basados en los principios de la selección y la evolución natural. Las posibles soluciones a un problema son codificadas en forma de cadenas binarias y la búsqueda se inicia con una población de posibles soluciones generada aleatoriamente. En dicha cadena, cada variable tiene asignada una posición o bits, de manera que una posible solución vendrá descrita por una sucesión de unos y ceros indicando la presencia (con un uno) o la ausencia (con un cero) de cada una de las variables en esa combinación particular. En las condiciones genéticas cada variable es llamada gen y un juego de de variables es llamado cromosoma.

    En este tipo de algoritmos, cada miembro de la población, que representa una posible solución, es testeada con algún criterio objetivo de manera de cada uno de los miembros de la población se valora en función de su valor del criterio. A las soluciones mejor valoradas se les permite sobrevivir y pasar a la siguiente iteración (“generación”), mientras que las soluciones con peor valoración desaparecen en sucesivas generaciones. El algoritmo genético prosigue hasta que iguala o supera el valor de criterio establecido como meta, hasta que exista una convergencia en la población, de manera que un determinado porcentaje de sus miembro acaben siendo idénticos o hasta que se llegue al número máximo de iteraciones.

**2.- Funciones de Base Radial (RBF)**

Una función de base radial presenta las siguientes ventajas [22]:

- Usa un número mínimo de nodos en comparación con otras redes, reduciendo el número de cálculos para el aprendizaje.
- La arquitectura es simple (Solo 3 capas).
- Tienen amplia eficiencia en la fase de entrenamiento.
- Mejor desempeño con un mayor número de datos de entrenamiento.
- Varios algoritmos pueden ser empleados para encontrar los parámetros más apropiados para la función de base radial
- Buen clasificador de patrones.

La función de base radial se puede definir de manera matemática como sigue:

$$F(x) = \sum_{i=1}^{N} \mu_i \cdot\cdot \left( \| x - c_i \| \right) \qquad \text{(Ec 37)}$$

Donde $\left\{ \cdot\cdot \left( \| x - c_i \| \right) | i = 1,2,..., N \right\}$ es un conjunto de funciones, generalmente no lineales, conocidas como funciones radiales básicas, y denota una norma que usualmente es Euclidiana. $\mu_i$ denota un factor de peso, $N$ es el número de funciones radiales de la red $F(x)$ es la salida que proporciona la red.

Básicamente, la función de base radial se constituye de 3 capas: Una capa de entrada, encargada de recibir los datos que se van a procesar. Esta capa puede estar constituida por sensores que adecuan las señales para que las pueda recibir la red neuronal. La segunda capa, o capa oculta, es la que se encarga de procesar los datos de acuerdo al tipo de función matemática que se tiene, como por ejemplo la función en ecuación (24), y una última capa de salida que es la que entrega el resultado que proporciona la red para la entrada dada.

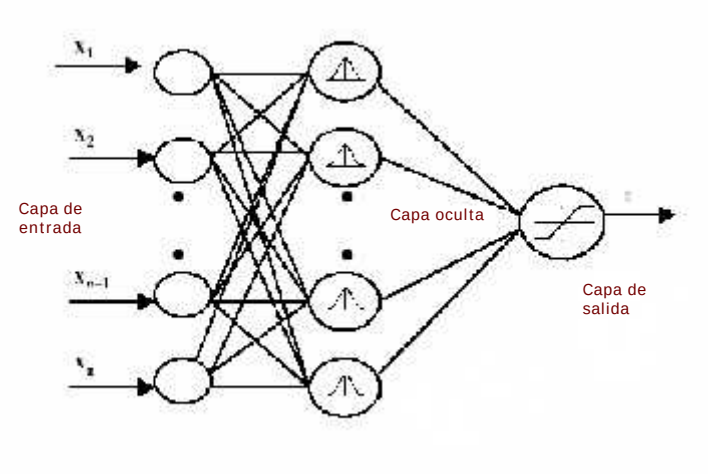


Figura 10: Capas de una RBF

Los nodos ocultos, contienen 2 parámetros principales, que son:

- Centro. Existe un centro para cada función radial involucrada en la capa oculta de la red, y es el punto en donde la función tiene un valor máximo.
- Ancho. Es el término usado para identificar a la amplitud de la función radial identificada por la campana de Gauss.

En la figura 11 se muestra una función (neurona) Gaussiana, en donde se pueden apreciar el centro y el ancho se la misma

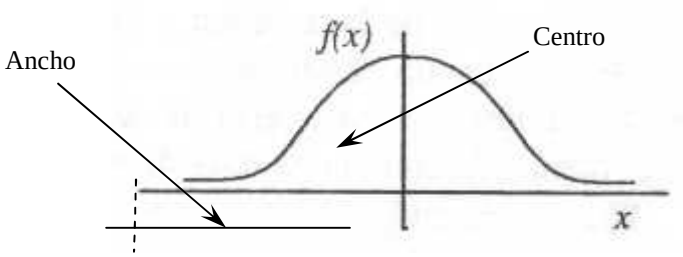


Figura 11: Función Gaussiana

Primeramente, ingresa un vector de entrada *x* a través de la primera capa. Posteriormente, en la capa oculta se busca la distancia radial entre el centro de cada función Gaussiana y el vector de entrada *x*, con "n" observaciones,

$$d = \|x - c\| = \sqrt{\sum_{i=1}^{n} (x_i - c_i)^2} \qquad \text{(Ec 38)}$$

El valor que se obtiene de la Ecuación (25) es una componente de entrada utilizada para activar la función radial.

En la capa oculta, se puede observar que en la medida que los valores de entrada se aproximen más a su centro la distancia tenderá a cero y de este modo la función Gaussiana se dispara a las vecindades de uno. Por otro lado, en la medida que los valores de entrada se alejen de su centro la distancia será mayor y la función radial tendería a cero. Este proceso corresponde a una clasificación no lineal de las entradas.

En una función de base radial se usa un proceso de entrenamiento híbrido, que se compone principalmente de fase supervisada y fase no supervisada. El entrenamiento de la función de base radial ayuda a determinar los siguientes parámetros:

- Centro y ancho: la determinación de estos parámetros se realiza mediante la optimización en el espacio de entradas, ya que cada neurona va a representar una zona diferente en dicho espacio. Esta parte del entrenamiento corresponde a la fase no supervisada.
- Pesos: para la determinación de los pesos, la optimización se realiza en base a las salidas que se desea obtener. Esta parte corresponde a la fase supervisada del entrenamiento. Los pesos se pueden calcular mediante técnicas de minimización de error.

**3.- Cortafuego**

Un Cortafuego es un sistema que protege a un computador o a una red de computadores de posibles intrusos provenientes de redes externas (generalmente desde Internet). Su función principal es obligar a que todas las conexiones que se establezcan entre el interior y el exterior de una red protegida pasen a través de él, que hará de intermediario, para ser inspeccionadas y permitidas o denegadas conforme a la política de seguridad establecida. Entre las tareas que desempeña un cortafuego, se puede destacar [1]:

- Protección contra ataques basados en el encaminamiento de paquetes IP.
- Protección contra servicios indeseados.
- Control de acceso a los recursos de la red.
- Monitorización de eventos relacionados con la seguridad de la red. El cortafuego puede realizar un registro detallado de todos los eventos ocurridos, de forma que sean posteriormente analizados.
- Centralización y control de la política de seguridad.
- Funciones no estrictamente relacionadas con la seguridad. Puede realizar tareas de traducción de direcciones locales a direcciones de Internet.

Las decisiones de filtrado están basados en el análisis de algunos de los campos presentes en la estructura del paquete. Si se trata de paquetes IP, se basará en los campos de la cabecera, como son las direcciones IP de origen y destino. Además, en algunos casos, el filtrado se realiza también basándose en la cabecera a nivel de capa de transporte del modelo OSI, analizando el puerto TCP o UDP de origen y/o destino, los cuales sirven, a su vez, para conocer el tipo de aplicación de que se trata [1].

En la figura 12 se representa una configuración de cortafuego:

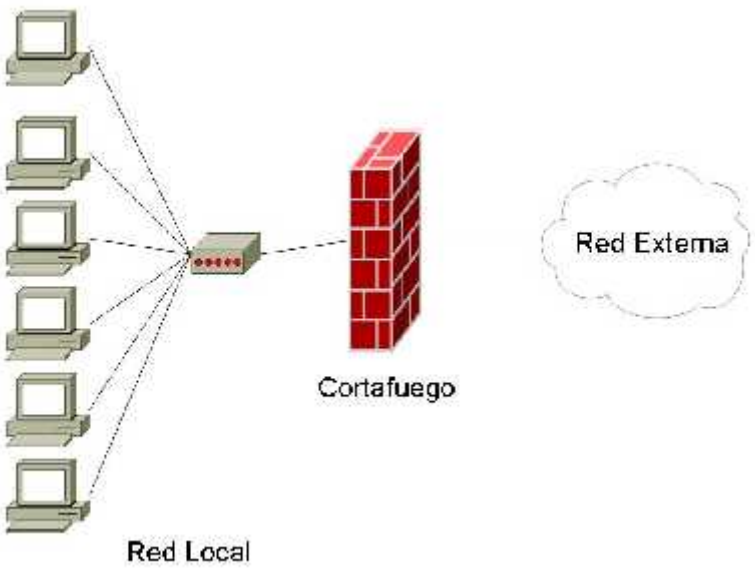


Figura 12: Representación de la implantación de un cortafuego en una red

Los Cortafuego, por lo general, son muy efectivos en bloquear el acceso a servicios protegidos, servicios públicos bien conocidos. Estos Cortafuego por naturaleza son una defensa del perímetro, y una vez que ha sido penetrado, la red interna quedará desprotegida [27].

Un Cortafuego normalmente tiene problemas en distinguir a nivel de aplicación, entre un tráfico normal dirigido a un servicio público de uno que corresponda a un ataque, implementar esta característica en un Cortafuego significará un incremento en la carga de procesamiento de data, aumentando con ello el cuello de botella que significa tener una única vía para el tráfico de entrada y salida de una red [27].

**4.- Sistema de Detección de Intrusos**

Un IDS es aquel que permite recabar información de distintas fuentes del sistema en el que se implanta para alertar de un posible ataque en las redes o computadores. La alerta puede ser del hecho de que existe un intento de ataque, así como del modo en el que este se está realizando y en algunos casos por parte de quién esta siendo efectuado. Se puede considerar un sistema de detección de intrusos como una herramienta de control que permitirá tomar decisiones a la hora de realizar una auditoria de seguridad del sistema computacional.

Se puede definir intrusión como la violación de la política de seguridad de un sistema, o como la materialización de una amenaza, la cual corresponde a un conjunto de acciones que tratan de comprometer la integridad, confidencialidad o disponibilidad de un recurso. Una de las definiciones más populares de intrusión es: fallo operacional maligno, inducido externamente, aunque es bien sabido que muchas de las intrusiones proceden del interior del sistema de información [21].

**a.- Clasificación de los Sistemas de Detección de Intrusos**

La clasificación más común se realiza en base a tres características funcionales de los IDS:

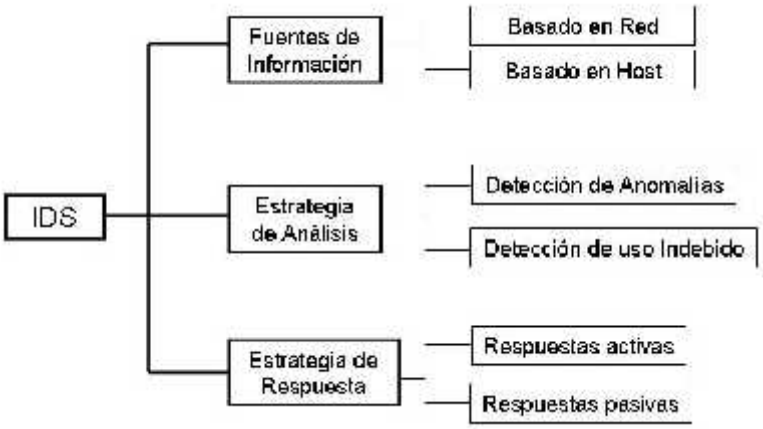


Figura 13: Clasificación de los IDS [21].

- **Fuentes de información.**

  Se refiere al origen de los datos que se usan para determinar si una intrusión se ha llevado a cabo [21].

  Desde el inicio de los IDS, se ha trabajado con multitud de datos provenientes de diferentes fuentes para tratar de identificar la existencia de una intrusión. Estos datos se pueden diferenciar en dos grandes grupos; aquellos datos obtenidos de una máquina o host, y aquellos datos obtenidos a partir de la monitorización de una red. Dentro de cada grupo, se pueden identificar diferentes enfoques que se pueden tomar, algunos ejemplos son los siguientes:

    - Host:
      - Logs o registros de auditoria
      - Llamadas del sistema de procesos en ejecución
      - Métricas de uso del sistema
      - Comandos del teclado

    - Red:
      - Comunicación de datos (Ethernet, Token Ring, …)
      - Nivel de red (normalmente IP)
      - Nivel de Transporte/control (TCP, UDP, RTP, ICMP,…)
      - Nivel de Aplicación (HTTP, DNS, Telnet, FTP, SSH, SMTP,…)

- **Estrategia de Análisis.**

  Se trata del método de detección utilizado. La información recogida en el paso anterior puede ser analizada mediante diferentes estrategias [21].

  i) Detección de uso indebido

  Un IDS basado en detección de uso indebido monitorea las actividades que ocurren en un sistema y las compara con firmas de ataques, las cuales se encuentran almacenadas en una base de datos. Cuando las actividades monitoreadas coinciden con las firmas, genera una alarma. La detección de intrusos basada en uso indebido se atiene al conocimiento a priori de las secuencias y actividades que forman un ataque. Con este método se detectan las tentativas de explotación de vulnerabilidades conocidas o patrones de ataque típicos. Esta estrategia es la más utilizada en los Sistemas de Detección de Intrusos comerciales [3].

Típicamente, un sistema de detección de uso indebido contiene dos componentes principales:

- Un lenguaje o modelo para describir o representar las técnicas utilizadas por los atacantes.
- Programas de monitoreo para detectar la presencia de un ataque basado en las representaciones o descripciones dadas.

La ventaja de los Sistemas de Detección de Intrusos basados en uso indebido es la fidedigna detección de patrones de ataques conocidos. Al igual que un software antivirus, el comportamiento malévolo puede identificarse con una precisión aceptable.

Como desventaja, cabe mencionar el hecho de que el patrón del ataque ha de ser conocido con anterioridad, lo que hace que nuevas intrusiones pasen desapercibidas ante el detector, o que el sistema pueda ser fácilmente engañado con pequeñas variantes de los patrones de ataques conocidos. Otra desventaja es que hay que adaptar manualmente el Sistemas de Detección de Intrusos a la plataforma en el que se implanta si no se quiere que se dispare el número de falsos positivos [21].

Se pueden encontrar diferentes métodos para implementar dichos sistemas:

- Sistemas basados en conocimiento

  Este tipo de métodos comprueban los eventos de hosts o redes en busca de reglas o patrones de ataque predefinidos. El objetivo es emplear representaciones de ataques suficientemente conocidos para el manejo de ocurrencias de dichos ataques. El modo de representar los ataques se ha realizado mediante sistemas expertos, firmas de ataques, transición de estados, y también se puede encontrar un caso particular de redes de Petri [21].

  - Sistemas Expertos. Dichos sistemas codifican el conocimiento en bases de datos mediante reglas de implicación (condición-acción). Cuando se cumplen todas las condiciones, la regla se activa y se ejecuta la consecuencia de la regla. El motor de inferencia será el encargado de decidir si ha ocurrido una intrusión haciendo uso de las reglas y los hechos. La ventaja de este sistema es el de la separación de la lógica de control sobre el dominio del problema; pero tiene la limitación de que básicamente las reglas no son temporalmente secuenciales, lo que hace difícil especificar pasos de intrusiones basadas en el tiempo [21].

- Detección de Firmas. Esta variante, también conocida como Sistema de Razonamiento Basado en Modelos, observa la ocurrencia de cadenas especiales (o patrones de cadenas) que puedan ser consideradas como sospechosas. La detección de firmas compara los eventos que ocurren, con las cadenas o firmas almacenadas en una base de datos de escenarios de ataque (almacenada como una secuencia de comportamientos o actividades) en busca de coincidencias. Su principal inconveniente es la necesidad de desarrollar e incorporar a la base de datos una firma nueva para cada nuevo tipo de ataque o vulnerabilidad descubierta [3].

- Análisis de Transición de Estados. Se crean a partir de la construcción de una máquina de estados finitos. Los escenarios de ataques se representan como una secuencia de transiciones que caracterizan la evolución del estado de seguridad de un sistema. Cuando el autómata alcanza un estado considerado como una intrusión, se lanza la alarma. Sus ventajas son la simplicidad conceptual, generalidad, y representación gráfica. Sin embargo, buscar equivalencias entre una red compleja y los eventos de auditoria puede resultar computacionalmente caro [3].

- Sistemas basados en Aprendizaje Automático

  Los métodos de aprendizaje automático o Machine Learning utilizados para la detección de uso indebido, descubren o generan patrones y clases de ataques automáticamente, en lugar de generarlos de forma manual o predefinida. Dichos métodos explotan las regularidades o asociaciones inherentes en los datos. El objetivo sigue siendo el de crear representaciones de ataques, con la diferencia de que éstas se inducen de forma automática, evitando así el diseño costoso de las representaciones vistas anteriormente. Para ello, el sistema realiza la fase de aprendizaje a partir de secuencias conocidas de ataques para obtener patrones de intrusiones [21].

  Para la detección de uso indebido se realiza la construcción de modelos predictivos etiquetados (como “normal”/“intrusivo”) [21].

ii) Detección de anomalías

Una anomalía se puede definir como la discrepancia de una regla o de un uso. De ese modo, el primer paso de un sistema de detección de anomalías comienza por establecer lo que se considera comportamiento normal de un sistema (usuarios, redes, registros de auditoria, llamadas del sistema de los procesos). Una vez definido esto, clasificará como sospechosas o intrusivas aquellas desviaciones que se puedan detectar sobre el comportamiento normal [21].

La detección de anomalías depende mucho de la suposición de que los usuarios y las redes se comportan de un modo suficientemente regular, de forma que cualquier desviación significante pueda ser considerada como evidencia de una intrusión [3].

La gran ventaja de la detección de anomalías es que el sistema es capaz de aprender el comportamiento normal del objeto de estudio, y a partir de ahí detectar desviaciones del mismo, clasificándolas como intrusiones. De este modo, se demuestra que es capaz de detectar tipos de ataques hasta el momento desconocidos [21].

Como desventaja, por definición únicamente señala comportamientos inusuales, pero éstos no tienen necesariamente por qué ser ilícitos. Por ello, destaca el problema de su alta tasa de falsos positivos. Otra desventaja de este proceso es la falta de claridad (es un proceso borroso). Un intruso podría actuar lentamente y realizar sus acciones cuidadosamente para modificar el perfil de los usuarios de modo que sus actividades serían aceptadas como legales cuando en realidad deberían lanzar una alarma (falsos negativos). Otras veces, no es o debería ser suficiente el hecho de simplemente avisar de un comportamiento anómalo sin explicar los posibles orígenes [21].

Al igual que ocurre con la detección de uso indebido, se pueden encontrar diferentes variantes en el método de implementar los sistemas de detección de anomalías. Se hacen uso de mecanismos heurísticos y estadísticos para adaptarse a los cambios en el comportamiento del objeto a estudio así como para detectar cambios imprevistos. Otras aproximaciones tratan de incorporar otras técnicas para realizar esta función [21].

- Sistemas Basados en Conocimiento
    - Sistemas Expertos

      Los sistemas expertos fueron inicialmente los más utilizados para la realización de IDS. Dentro de esta categoría existen modelos que se basan en la hipótesis de que las violaciones de seguridad pueden detectarse mediante la monitorización de los registros de auditoria del sistema en busca de patrones de uso del sistema anormales, y utiliza reglas para la adquisición de conocimiento a partir de registros de auditorias. Un segundo modelo analiza los datos de auditoria recogidos de varios sistemas interconectados en búsqueda de actividades que puedan indicar comportamiento inusual y/o malicioso de los usuarios. El análisis se realiza mediante dos unidades complementarias de detección: el subsistema de análisis de firmas basado en reglas, y el subsistema estadístico de detección de anomalías basado en perfiles generados [21].

- Sistemas Basados en Métodos Estadísticos

  Dichos sistemas incluyen perfiles para representar el comportamiento de los sujetos con respecto a los objetos en términos de métricas y modelos estadísticos, y reglas para la adquisición de conocimiento a partir de registros de auditorias y también para la detección del comportamiento anómalo. Un componente básico de dichos sistemas es el de los perfiles de actividades, los cuales caracterizan el comportamiento de un sujeto (normalmente usuarios) con respecto a un objeto (archivos, programas, registros). Dicha caracterización se realiza mediante el establecimiento de métricas, y modelos estadísticos (como modelo operacional, modelo de significancia y desviación estándar, modelo multivariado, modelo del proceso de Marcov y modelo de series temporales) [21].

- Sistemas Basados en Aprendizaje Automático

  Este tipo de sistemas ha sido y sigue siendo el más estudiado como método para el modelado de comportamientos normales. Debido a la gran variedad de modelos de aprendizaje automático existentes, se han realizado muchas pruebas tratando de dar con el modelo que mejores resultados aporte en cuanto a su precisión de detección, reducción de falsos positivos y tiempo de computación. En varios trabajos se han unido diferentes modelos, tanto a modo comparativo, como tratando de utilizar aquel modelo que mejor se ajuste para cada caso (tipo de ataque) [21].

- **Respuestas:**

  Una vez que se ha determinado si ha sucedido alguna intrusión, los IDS pueden o bien responder de forma activa ante la misma, o bien registrar la detección y no realizar acción alguna [21].

  La gran mayoría de los Sistemas de Detección de Intrusos cuentan con un método de respuesta básico cuando identifican un ataque: la notificación. A este tipo de respuesta se le llama respuesta pasiva, y su función es la de notificar al administrador de la ocurrencia de un ataque. La notificación suele realizarse por medio de mensajes, correo electrónico, SMS [21].

  Sin embargo, en los últimos años a tomado fuerza la posibilidad de responder a los ataques de forma automática, son las llamadas respuestas activas. Este tipo de respuestas son un campo activo de investigación, debido a que por un lado, las respuestas que se implementan hoy en día ignoran el costo que puede suponer una intrusión. De este modo podría ocurrir que las respuestas causaran mayor daño que las propias intrusiones. Por el otro lado, los Sistemas de Detección de Intrusos actuales reportan un gran número de falsos positivos, por lo que pueden causar acciones de respuesta numerosas, innecesarias y costosas pudiendo llegar a causar denegación de servicio a usuarios legítimos del sistema [21].

## 5.- Tipos de Ataques

En general se pueden dividir en cuatro clases:

### a) Probing

Es una clase de ataque en la cual un atacante escanea una red para recabar información o encontrar posibles vulnerabilidades conocidas. Un atacante con información de las máquinas y servicios que están disponibles en una red puede usarla para explotar algunas vulnerabilidades conocidas.

Existen diferentes tipos de ataques de sondeo: como el abuso de características legítimas de las computadoras, o el uso de técnicas de ingeniería social. Por otro lado, esta clase de ataques son los más escuchados y requiere muy poca experiencia técnica para llevarlos a cabo.

Diferentes tipos de ataques de sondeo se ilustran en la siguiente tabla:

| **TIPO DE ATAQUE** | **SERVICIO** | **MECANISMO** | **EFECTO DEL ATAQUE** |
|---|---|---|---|
| IPSWEEP | ICMP | Abuso de características | Identifica Máquinas Activas |
| MSCAN | Muchos tipos | Abuso de características | Busca vulnerabilidades conocidas |
| NMAP | Muchos tipos | Abuso de características | Identifica puertos activos en una máquina |
| SAINT | Muchos tipos | Abuso de características | Busca vulnerabilidades conocidas |

Tabla 1: Ejemplo de ataques de sondeo (Probing)

**b) Ataque de Denegación de Servicio (DoS)**

Denegación de Servicio (DoS) es una clase de ataques en la cual un atacante hace que los recursos de cómputo o de memoria se encuentren demasiado ocupados para manejar peticiones legítimas de servicio, logrando negar a los usuarios legítimos el acceso a una máquina o servicio.

Existen diferentes formas de lanzar ataques de DoS: al abusar de las funciones legítimas de los computadores, explotando alguna vulnerabilidad de una aplicación, o mediante la explotación de errores del sistema.

Los ataques de DoS se clasifican sobre la base de los servicios impedidos de acceder por parte de los usuarios legítimos. Algunos de este tipo de ataque se ilustran en la siguiente tabla:

| **TIPO DE ATAQUE** | **SERVICIO** | **MECANISMO** | **EFECTO DEL ATAQUE** |
|---|---|---|---|
| APACHE2 | http | Abuso | Detiene servicio httpd |
| BACK | http | Abuso/Vulnerabilidad | Respuesta más lenta del servidor |
| LAND | http | Vulnerabilidad | Hiberna la máquina |
| MAIL BOMB | N/A | Abuso | Molestia |
| SYN FLOOD | TCP | Abuso | Niega servicio en uno o más puertos |
| PING OF DEATH | ICMP | Vulnerabilidad | Nada |
| PROCESS TABLE | TCP | Abuso | Niega nuevos procesos |
| SMURF | ICMP | Abuso | Respuesta más lenta de la red |
| SYSLOGD | Syslog | Vulnerabilidad | Detiene syslogd |
| TEARDROP | N/A | Vulnerabilidad | Reinicia la máquina |
| UDPSTROM | Echo/Chargen | Abuso | Respuesta más lenta de la red |

Tabla 2: Ejemplos de ataques de denegación de servicios

**c) User to Root**

Este tipo de ataque usualmente se inicia con el acceso del atacante a través de una cuenta de usuario normal, y aprovechando algunas vulnerabilidades logra acceder como administrador de la máquina.

En el siguiente cuadro se presentan algunos tipos de ataque de esta categoría.

| **TIPO DE ATAQUE** | **SERVICIO** | **MECANISMO** | **EFECTO DEL ATAQUE** |
|---|---|---|---|
| EJEC. | Sesión de Usuario | desbordamientos de búfer | Acceder a una shell como Root |
| FFBCONFIG | Sesión de Usuario | desbordamientos de búfer | Acceder a una shell como Root |
| FDFORMAT | Sesión de Usuario | desbordamientos de búfer | Acceder a una shell como Root |
| XTERM | Sesión de Usuario | desbordamientos de búfer | Acceder a una shell como Root |

Tabla 3: Ejemplo de Ataque User to Root

**d) Ataque de remoto a usuario**

Corresponde a una clase de ataque donde el atacante envía mensajes a una máquina atravesando la red, logrando explotar las vulnerabilidades de la máquina y logrando un acceso a esta como usuario local en forma ilegal. Algunos ataques de este tipo se presentan a continuación:

| TIPO DE ATAQUE | SERVICIO | MECANISMO | EFECTO DEL ATAQUE |
|---|---|---|---|
| DICTIONARY | TELNET, RLOGIN, POP, FTP, IMAP | Abuso | Acceder a una shell como Root |
| FTP-WRITE | FTP | Error de Configuración | Acceder a una shell como Root |
| GUEST | TELNET, RLOGIN | Error de Configuración | Acceder a una shell como Root |
| IMAP | IMAP | Vulnerabilidad | Acceder a una shell como Root |
| NAMED | DNS | Vulnerabilidad | Acceder a una shell como Root |
| PHF | HTTP | Vulnerabilidad | Ejecutar comandos como usuario http |
| SENDMAIL | SMTP | Vulnerabilidad | Ejecutar comandos como usuario Root |

Tabla 4: Ejemplos de ataque de Remoto a Usuario

**6.- Conceptos de los Protocolos Analizados**

Un protocolo es un método establecido de intercambiar datos entre dos computadores, a través de este método los computadores elijen como comunicarse.

El protocolo determinará lo siguiente:

- El tipo de comprobación de errores que se utilizará.
- El método de compresión de los datos, si lo hay.
- Cómo indicará el dispositivo que envía que ha acabado de enviar un mensaje.
- Cómo indicará el dispositivo que recibe que ha recibido un mensaje.

Un servidor conectado a Internet principalmente trabajará con un rango limitado de protocolos, estos son: IP, ICMP, TCP y UDP. Por lo cual a continuación se describirán dichos protocolos y los campos que componen los encabezamientos de cada uno de estos.

**a) Protocolo IP:**

El Protocolo IP proporciona un sistema de distribución que es poco fiable. El protocolo IP especifica que la unidad básica de transferencia de datos es el datagrama. Los datagramas pueden ser retrasados, perdidos, duplicados, enviados en una secuencia incorrecta o fragmentada intencionalmente para permitir que un nodo con un buffer limitado pueda coger todo el datagrama. Es la responsabilidad del protocolo IP reensamblar los fragmentos del datagrama en el orden correcto. Cuando los datagramas viajan de unos equipos a otros, es posible que atraviesen diferentes tipos de redes. El tamaño máximo de estos paquetes de datos puede variar de una red a otra, dependiendo del medio físico que se emplee para su transmisión. A este tamaño máximo se le denomina MTU (Maximum Transmission Unit), y ninguna red puede transmitir un paquete de tamaño mayor a esta MTU. El datagrama consiste en una cabecera y datos.

Encabezado:


Figura 14: Campos de Encabezamiento IP

- Longitud de la Cabecera: Este campo ocupa 4 bits, y representa el número de octetos de la cabecera dividido por cuatro, lo que hace que este sea el número de grupos de 4 octetos en la cabecera.
- Versión: El campo versión ocupa 4 bits. Este campo hace que diferentes versiones del protocolo IP puedan operar en la Internet. En este caso se trata de la versión 4.
- Tipo de servicio: Este campo ocupa un octeto de la cabecera IP, y especifica la precedencia y la prioridad del datagrama IP. Los tres primeros bits del octeto indican la precedencia. Los valores de la precedencia pueden ser de 0 a 7. Cero es la precedencia normal, y 7 está reservado para control de red. Los otros 4 bits definen el campo prioridad, que tiene un rango de 0 a 15. Las cuatro prioridades que están asignadas son: 0, (por defecto, servicio normal), 1 (minimizar el coste monetario), 2 (máxima fiabilidad), 4 (maximizar la transferencia), 8 (El bit +4 igual a 1, define minimizar el retraso). Estos valores son utilizados por los routers para direccionar las solicitudes de los usuarios.
- Longitud Total: Este campo se utiliza para identificar el número de octetos en el datagrama total.
- Identificación: El valor del campo identificación es un número secuencial asignado por el Host origen. El campo ocupa dos octetos. Los números oscilan entre 0 y 65.535, que cuando se combinan con la dirección del Host forman un número único en Internet. El número se usa para ayudar en el reensamblaje de los fragmentos de datagramas.
- Fragmentos Offset: Cuando el tamaño de un datagrama excede el MTU, este se segmenta. El fragmento Offset representa el desplazamiento de este segmento desde el inicio del datagrama entero.
- Banderas: El campo ocupa 3 bits y contiene dos Banderas. El valor 5 del campo se utiliza para indicar el último datagrama fragmentado cuando toma valor cero. El valor 7 lo utiliza el servidor origen para evitar la fragmentación. Cuando este bit toma un valor diferente de cero y la longitud de un datagrama excede el MTU, el datagrama es descartado y un mensaje de error es enviado al Host de origen por medio del protocolo ICMP.
- Tiempo de Vida: El campo tiempo de vida ocupa un octeto. Representa el número máximo de segundos que un datagrama puede existir en Internet, antes de ser descartado. Un Datagrama puede existir un máximo de 255 segundos. El número recomendado para IP es 64. El originador del datagrama manda un mensaje ICMP cuando el datagrama es descartado.
- Protocolo: El campo protocolo se utiliza para identificar la capa de mayor nivel más cercana usando el IP. Este es un campo de 0 bits, que normalmente identifica tanto la capa TCP (valor 6), como la capa UDP (valor 17) en el nivel de transporte, pero puede identificar hasta 255 protocolos de la capa de transporte.
- Checksum: El checksum proporciona la seguridad de que el datagrama no ha sido dañado ni modificado. Este campo tiene una longitud de 16 bits. El checksum incluye todos los campos de la cabecera IP, incluido él mismo, cuyo valor es cero a efectos de cálculo.
- Dirección de Origen: Este campo contiene un identificador de red (Netid) y un identificador de Host (Hostid). El campo tiene una longitud de 32 bits

- Dirección de Destino: Este campo contiene el Netid y el Hostid del destino. El campo tiene una longitud de 32 bits.
- Opciones: La existencia de este campo viene determinada por la longitud de la cabecera. Si esta es mayor de cinco, por lo menos existe una opción. Aunque un Host no está obligado a poner opciones, puede aceptar y procesar opciones recibidas en un datagrama. El campo Opciones es de longitud variable. Cada octeto esta formado por los campos Copia, Clase de Opción y Número de Opción.
    - El campo Copia sirve para que cuando un datagrama va a ser fragmentado y viaja a través de nodos o Gateways. Cuando tiene valor 1, las opciones son las mismas para todos los fragmentos, pero si toma valor 0, las opciones son eliminadas.
    - Clase de Opción es un campo que cuando tiene valor 0, indica datagrama o control de red; Cuando tiene valor 2, indica depuración o medida. Los valores 1 y 3 están reservados para un uso futuro.
    - El Número de Opción indica una acción específica.

**b) Protocolo ICMP**

Internet es un sistema autónomo que no dispone de ningún control central. El protocolo ICMP (Internet Control Message Protocol), proporciona el medio para que el software de hosts y Gateways intermedios se comuniquen. El protocolo ICMP tiene su propio numero de protocolo (numero 1), que lo habilita para utilizar el protocolo IP directamente. La implementación de ICMP es obligatoria como un subconjunto lógico del protocolo IP. Los mensajes de error de este protocolo los genera y procesa TCP/IP, y no el usuario.

Encabezado:

- Tipo: Puede tener los siguientes valores según se trate el tipo de mensaje:
    - 0 Respuesta de Eco: Un Host puede comprobar si otro Host se encuentra operativo mandando una solicitud de eco. El receptor de la solicitud la devuelve a su origen. Esta aplicación recibe el nombre de Ping. Esta utilidad encapsula la solicitud de eco del ICMP (tipo 8) en un datagrama IP y lo manda a la dirección IP. El receptor de la solicitud de eco intercambia las direcciones del datagrama IP, cambia el código a 0 y lo devuelve al origen.


Figura 15: Campos encabezamiento Mensaje Echo ICMP

- 3 Destino Inalcanzable: Si un Gateways no puede enviar un datagrama a la dirección de destino, este manda un mensaje de error ICMP al origen. El valor del campo tipo es 3, y el tipo de error viene dado por el campo código
  - 0 Red no alcanzable
  - 1 Host no alcanzable
  - 2 Protocolo no alcanzable
  - 3 Puerto no alcanzable
  - 4 Necesaria fragmentación con la opción DF
  - 5 Fallo de la ruta de origen
  - 6 Red de Destino desconocida
  - 7 Host de Destino desconocido
  - 8 Fallo del Host de Origen
  - 9 Red prohibida administrativamente
  - 10 Host prohibido administrativamente
  - 11 Tipo de servicio de Red no alcanzable
  - 12 Tipo de servicio de Host no alcanzable.

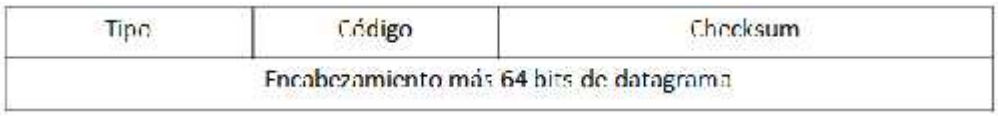


Figura 16: Campos encabezamiento Mensaje Destino Inalcanzable ICMP

- 4 Origen saturado: Para contener los datagramas IP, un Gateways dispone de un buffer. Si el número de datagramas es grande, el buffer se satura. En este momento el Gateways descarta todos los mensajes que recibe hasta que obtiene un nivel de buffer aceptable. Cada datagrama descartado hace que el Gateways mande un mensaje ICMP de control de flujo al origen. Esto informa de que un mensaje ha sido descartado. Originalmente el mensaje ICMP de control de flujo se enviaba cuando el buffer estaba lleno, pero esto llegaba demasiado tarde, y el sistema ya estaba saturado. El algoritmo se ha cambiado para que el mensaje ICMP de control de flujo se envíe cuando el buffer se encuentre al 50%.

  El formato del mensaje de control de flujo es idéntico al mensaje de inalcanzable, excepto que el tipo es 4 y el código es 0.

- 5 Redirección (cambiar ruta): Los Gateways en cualquier Internet contienen las tablas de redireccionamiento más comunes. Cuando la ruta por defecto no es la más adecuada, el Gateways puede enviar al Host un mensaje de redireccionamiento ICMP que contiene la ruta correcta.

  El formato del mensaje ICMP de control de flujo es igual al del mensaje de Inalcanzable, excepto que el tipo es 5 y el valor del código es variable entre 1 y 3. Los motivos para la redirección son:

  1 Por el Host
  2 Por el tipo de servicio y red
  3 Por el tipo de servicio y Host

- 8 Solicitud de eco

- 11 Tiempo excedido para un datagrama: Para prevenir bucles en la redirección, el datagrama IP contiene un tiempo de vida definido por el origen. A medida que cada Gateways procesa el datagrama, el valor del campo disminuye en una unidad. Posteriormente el Gateways verifica si el valor del campo es 0. Cuando se detecta un 0, el Gateways manda un mensaje de error ICMP y descarta el datagrama.

  El formato del mensaje de error es igual al del mensaje de inalcanzable, pero el tipo es 11, y el código es igual a 0 (contador sobrepasado), o 1 (tiempo de ensamblaje de fragmento excedido).

- 13 Problema de parámetros en un datagrama: Un error de parámetros se produce cuando el que origina el datagrama, lo construye mal, o el datagrama está dañado. Si un Gateways encuentra un error en un datagrama, manda un mensaje ICMP de error de parámetros al origen y descarta el datagrama.

  El formato del mensaje ICMP de error de parámetros es igual al de inalcanzable, pero su tipo es 12, y el código es 0 si se utilizan punteros, o 1 si no se utilizan.

- 13 Solicitud de fecha y hora.

- 14 Respuesta de fecha y hora: El Mensaje Fecha y hora del ICMP es una herramienta útil para diagnosticar problemas de internet, y recoger información acerca del rendimiento. El protocolo NTP (Network Time Protocol), puede utilizarse para marcar el tiempo inicial, y puede guardar la sincronización en milisegundos del reloj.

  El mensaje Fecha y hora tiene los siguientes campos: Tipo, Código, Checksum, Identificador, Número de secuencia, Fecha y hora original, Fecha y hora receptor y Fecha y hora de transmisión. El tipo es igual 13 para el origen y 14 para el Host remoto. El código es igual a cero. El identificador y el número de secuencia se usan para identificar la respuesta. La Fecha y hora original es el tiempo en el que el emisor inicia la transmisión, la Fecha y hora receptor es el tiempo inicial en el que el receptor recibe el mensaje. La Fecha y hora de transmisión es el tiempo en que el receptor inicia el retorno del mensaje.

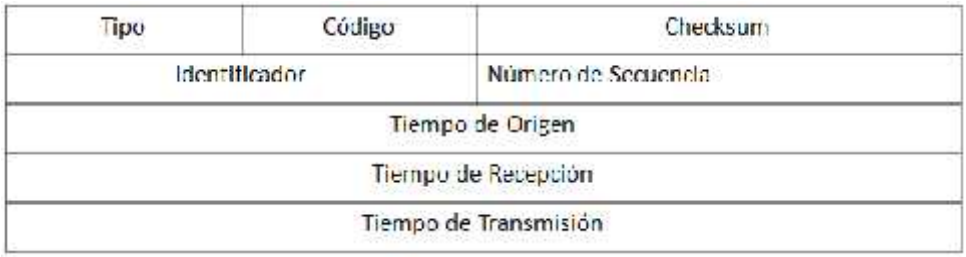


Figura 17: Campos encabezamiento Mensaje Fecha y Hora ICMP

- 17 Solicitud de mascara de dirección.

- 18 Respuesta de mascara de dirección: Cuando un Host quiere conocer la máscara de subred de una LAN física, puede mandar una solicitud ICMP de mascara de subred. El formato es igual a los primeros ocho octetos del ICMP Fecha y hora. El valor del campo tipo es 17 para la solicitud de mascara de subred y 18 para la respuesta. El código es 0, y el identificador y el número de secuencia se utilizan para identificar la respuesta.

**c) Protocolo TCP**

El protocolo TCP proporciona un servicio de comunicación que forma un circuito, es decir, que el flujo de datos entre el origen y el destino parece que sea continuo. TCP proporciona un circuito virtual el cual es llamado una conexión. Al contrario que los programas que utilizan UDP, los que utilizan TCP tienen un servicio de conexión entre los programas llamados y los que llaman, chequeo de errores, control de flujo y capacidad de interrupción.

**Interfaces TCP**

Existen dos tipos de interfaces entre la conexión TCP y los otros programas. El primero es utilizar la pila de los programas de la capa de red. Como en esta capa solo está el protocolo IP, la interface la determina este protocolo. El segundo tipo es la interfaz del programa de usuario. Esta interface puede variar según el sistema operativo, pero en general tiene las siguientes características:

- La interface envuelve el programa de usuario llamando a una rutina que introduce entradas en una estructura de datos llamada el bloque de control de transmisión.
- Las entradas se realizan inicialmente en la pila de hardware y transferidas al bloque de control de transmisión por medio de una rutina de sistema. Estas entradas permiten a TCP asociar un usuario con una conexión particular, de modo que pueda aceptar comandos de un usuario y mandarlos a otro usuario en el otro extremo de la conexión.
- TCP utiliza unos identificadores únicos para cada parte de la conexión. Esto se utiliza para recordar la asociación entre dos usuarios. Al usuario se le asigna un nombre de conexión para utilizarlo en una futura entrada del bloque de control de transmisión.
- Los identificadores para cada extremo de la conexión se llaman sockets. El socket local se construye concatenando la dirección IP de origen y el número de puerto de origen. El socket remoto se obtiene concatenando la dirección IP de destino y el número de puerto de destino.
- Un par de sockets (origen-remoto) de una conexión forman un único número en Internet. El protocolo UDP posee los mismos sockets, pero no los recuerda. Esta es la diferencia entre un protocolo orientado a conexión y otro no orientado a la conexión.

TCP recuerda el estado de cada conexión por medio del bloque de control de transmisión. Cuando se abre una conexión, se efectúa una entrada única en el bloque de control de transmisión. Cuando se cierra una conexión se elimina su entrada del bloque de control de transmisión.

**Control de Flujo**

El protocolo TCP puede controlar la cantidad de datos que debe enviar mediante el campo Window. Este campo indica el número máximo de octetos que pueden ser recibidos. El receptor de un segmento con el campo Window igual a cero, no puede enviar mensajes al emisor, excepto mensajes de prueba. Un mensaje de prueba es un mensaje de un solo octeto que se utiliza para detectar redes o hosts inalcanzables.

El segmento TCP consiste en una cabecera y datos.

Encabezado:


Figura 18: Campos encabezamiento Protocolo TCP

- Número de puerto del Origen/destino (Source/Destination Port Numbers): Este campo tiene una longitud de 16 bits.
- Números de Secuencia (Secuence Numbers): Existen dos números de secuencia en la cabecera TCP. El primer número de secuencia es el número de secuencia final (SSN). El SSN es un número de 32 bits. El otro número de secuencia es el Número de secuencia esperado de recepción, También llamado Número de Reconocimiento (acknowledgement number).
- Longitud de la cabecera (Header Length): Este campo tiene una longitud de 4 bits y contiene un entero igual al número de octetos que forman la cabecera TCP dividido por cuatro.
- Código de Bits (Code bits): El motivo y contenido del segmento TCP lo indica este campo. Este campo tiene una longitud de seis bits:
    - Bit URG (bit +5): Este bit identifica datos urgentes.
    - Bit ACK (bit +4): Cuando este bit se pone a 1, el campo reconocimiento es válido.
    - Bit RST (Bit +2): Poniendo este bit, se aborta la conexión. Todos los buffers asociados se vacíen.
    - Bit SYN (Bit +1): Este bit sirve para sincronizar los números de secuencia.
    - Bit FIN (Bit +0): Este bit se utiliza solo cuando se está cerrando la conexión.

- Ventana (Window): Este campo contiene un entero de 32 bits. Se utiliza para indicar el tamaño de buffer disponible que tiene el emisor para recibir datos.
- Opciones (Options): Este campo permite que una aplicación negocie durante la configuración de la conexión características como el tamaño máximo del segmento TCP. Si este campo tiene el primer octeto a cero, esto indica que no hay opciones.
- Relleno (Padding): Este campo consiste en un número de octetos (De uno a tres), que tienen valor cero y sirven para que la longitud de la cabecera sea divisible por cuatro.
- Checksum: Mientras que el protocolo IP no tiene ningún mecanismo para garantizar la integridad de los datos, ya que solo comprueba la cabecera del mensaje, TCP dispone de su propio método para garantizar dicha integridad. Como en el Checksum del protocolo TCP también se incluyen campos del protocolo IP, es necesario construir una pseudo-cabecera TCP que se considera únicamente para efectos de cálculo

**d) Protocolo UDP**

El protocolo UDP (User Datagram Protocol) proporciona aplicaciones con un tipo de servicio de datagramas orientado a transacciones. El servicio es muy parecido al protocolo IP en el sentido de que no es fiable y no está orientado a la conexión. UDP es simple, eficiente e ideal para aplicaciones de transferencia de video y voz. Una dirección IP sirve para dirigir el datagrama hacia una máquina en particular, y el número de puerto de destino en la cabecera UDP se utiliza para dirigir el datagrama UDP a un proceso especifico localizado en la cabecera IP. La cabecera UDP también contiene un número de puerto origen que permite al proceso recibido conocer cómo responder al datagrama

Encabezado:


Figura 19: Campos encabezamiento Protocolo UDP

- Números de Puerto de Origen y Destino: Estos números, junto con las direcciones IP definen el punto final de la comunicación. El número del puerto de origen, puede tener valor cero si no se usa. El número del puerto de destino solo tiene sentido en el contexto de un datagrama UDP y una dirección IP en particular. El número de puerto de origen es un campo de 16 bits. El puerto de destino tiene la misma longitud.

- Longitud del Mensaje: Este campo tiene una longitud de 16 bits y contiene el número total de octetos que forman el datagrama, incluída la cabecera.
- Checksum: El uso del checksum es opcional, y este campo debe ponerse a cero si no es utilizado. Mientras que el checksum del datagrama IP solo tiene en cuenta la cabecera del mensaje, UDP tiene su propio checksum para garantizar la integridad de los datos. La longitud de este campo es de 16 bits, y está formado por la suma de los campos de UDP, y algunos campos del protocolo IP.

# IX. PROPUESTA DE SOLUCIÓN

## 1.- DISEÑO

Una máquina que preste algún tipo de servicio tendrá que contar con al menos un cortafuegos, con el fin de minimizar el riesgo de sufrir un ataque. Esta asociación de funciones trabajará de la siguiente manera:

- Al recibir una consulta al servicio que presta la máquina, primeramente se compararán la dirección IP del emisor y el puerto al que se dirige la consulta con las reglas que posee el cortafuego. Si existiese alguna que prohíba la conexión se eliminará la consulta y se emitirá, si corresponde, un mensaje de respuesta. Si no existiese dicha regla entonces la consulta se enviaría al servicio correspondiente para ser procesada y respondida.

  En la figura 20 se grafica lo anteriormente expuesto:


Figura 20: Diagrama de flujos para una consulta básica en un servidor

Por lo cual, bajo esta premisa, el principal trabajo o realmente la única labor de seguridad se encuentra radicada en un cortafuego, que por lo general, será configurado con reglas estáticas y que por ende no es fácil de mantener actualizado, principalmente por la rápida evolución que tienen los diversos tipos de ataques. Al contar con un cortafuego cuyas reglas se encuentran desactualizadas o mal configuradas, el servicio se verá obligado a atender todo tipo de consultas, incluyendo posibles ataques.

Ahora bien, la tarea de mantener actualizada las reglas del Cortafuego puede llegar a convertirse en una labor extenuante para un administrador de red. Exigiendo de éste una cantidad elevada de tiempo en: administración del Cortafuego, constante análisis del tráfico que un servidor recibe y continuo estudio de nuevas técnicas que se deben implementar para los más recientes tipos de ataques que un servidor puede tener.

Como una manera de simplificar la administración de las actualizaciones de reglas de este cortafuego se propone diseñar un sistema que al detectar un tráfico lo pueda clasificar como anómalo o normal según lo aprendido anteriormente. Es así que al detectarse un tráfico clasificado como anómalo pueda anexar una regla que prohíba la conexión desde dicha dirección IP de origen. La figura 21 grafica lo anteriormente expuesto:

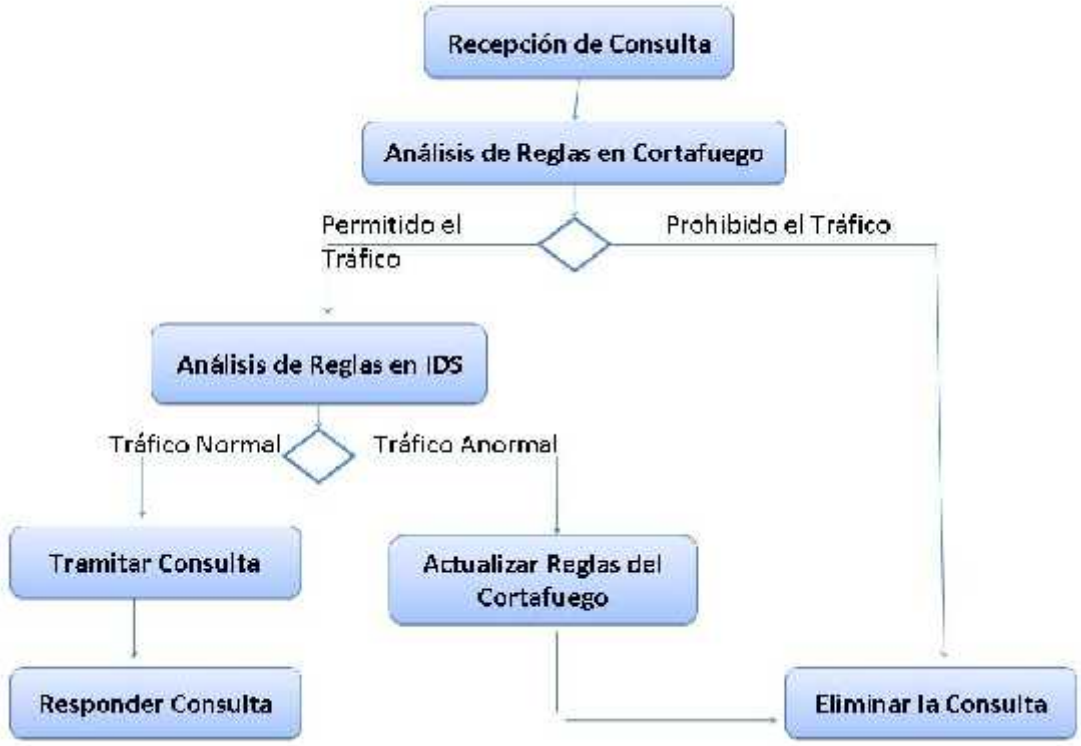


Figura 21: Diagrama de flujos para una consulta en un servidor con modificaciones

De la figura 21 se puede observar que las reglas del Cortafuego se actualizarán en forma automática, con lo cual se minimiza el trabajo del administrador en esta etapa, no requiere que el administrador tenga un gran conocimiento del tráfico que hay en la red y tampoco de los distintos tipos de ataques que van surgiendo. Por otro lado al ser automática la generación de las reglas, se logra evitar que el ataque se complete y que los usuarios detecten los efectos nocivos que conlleva dicho ataque.

Basado en el diagrama expuesto en la figura 21 se plantea la arquitectura del Cortafuego con un Sistema de Detección de Intrusos, que se ejemplifica en la figura 22:

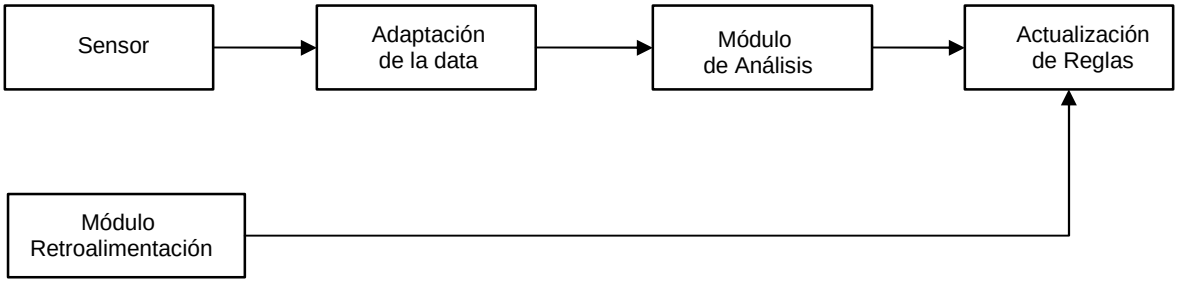


Figura 22: Arquitectura de Sistema Propuesto

Como se puede observar, el sistema está compuesto por los siguientes elementos:

- Sensor: el cual se encontrará monitoreando una posible consulta que sea recibida en el servidor, este monitoreo se realizará con la herramienta tcpdump; la cual imprime los encabezamientos de paquetes que son capturados a través de una interfaz de red, esta salida puede ser hecha a la pantalla o guardadas en un archivo determinado, para posterior análisis.

  La herramienta tcpdump capturará el primer paquete y lo guardará en un archivo temporal para ser analizado por los demás módulos. Una vez recibida esta, activará la adaptación de la data, y detendrá el monitoreo hasta que se termine el funcionamiento del módulo de análisis [6].

  Adicionalmente, en este módulo se utilizará la herramienta “inotifywait”; el cual corresponde a una herramienta disponible en Linux, que monitorea cualquier cambio en un archivo o directorio activando una acción previamente programada, es adecuada para ser utilizada en la programación Shell en Linux [7].

La figura 23, grafica lo anteriormente expuesto:

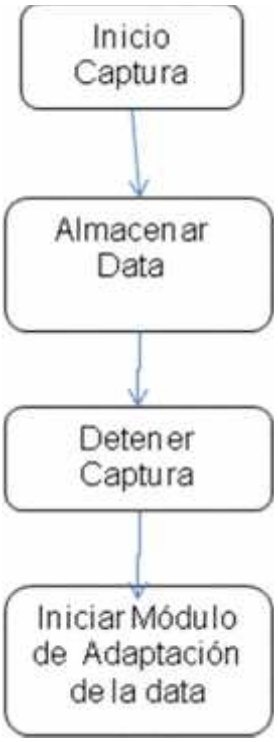


Figura 23: Esquema de tareas del Módulo Sensor

- Módulo Adaptación de la data: Una vez detectada la captura se procederá por un lado a transformar la data, en un formato requerido por el modulo de análisis, y se almacenará en un segundo archivo la dirección IP de origen que se podrá utilizar en la confección de las reglas del Cortafuego. Una vez realizada estás tareas se procederá a iniciar el Módulo de Análisis.

  El formato requerido por el módulo de análisis es:

  <label> <index1>:<value1> <index2>:<value2> ...

  En este punto se utilizará el lenguaje AWK, el cual es un lenguaje de programación que permite la manipulación de textos y archivos, es ampliamente utilizado en el desarrollo de utilidades de sistemas y tareas de administración [8].

La figura 24 representa las tareas que se desarrollan en el módulo de adaptación de data:

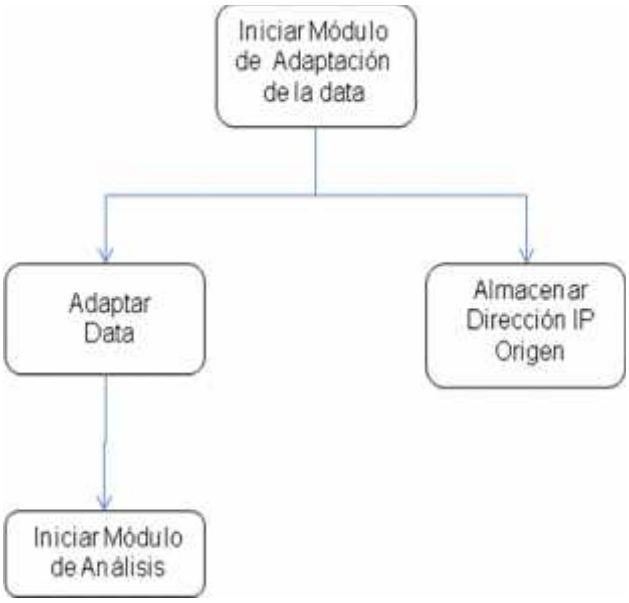


Figura 24: Esquema de tareas del Módulo de Adaptación de Data

- Módulo de Análisis: En este módulo se analizará el paquete capturado, y se marcará como un tráfico anómalo o normal, para realizar esta tarea se utilizarán las Máquinas De Soporte Vectorial, con la librería libsvm versión 2.89. Libsvm es una aplicación que soporta clasificación, regresión y estimación de distribución utilizando Máquinas De Soporte Vectorial [9] [13].

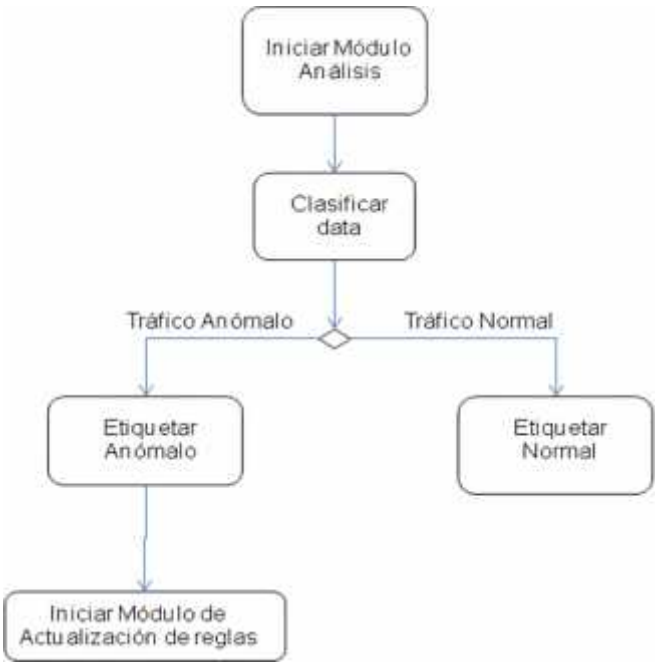


Figura 25: Esquema de tareas del Módulo de Análisis

- Módulo de Actualización de reglas: Para este trabajo se utilizará IPTABLES, el cual es una herramienta de Cortafuego, que permite: filtrar paquetes, traducciones de direccionamiento IP (NAT) y mantener registros de las actividades. Esta actualización se llevará acabo cuando se cumpla uno de las siguientes condiciones:

    - El módulo de análisis detecta un tráfico anómalo, con lo cual se inserta la siguiente regla en el archivo de configuración de IPTABLES:

      iptables -A INPUT -s direccion_IP_origen/32 -j DROP.

      Reiniciando el servicio para que tome efecto el cambio de configuración. Para lo anterior, se utilizará lenguaje de programación Shell.

    - El administrador desea deshacer un cambio en las reglas del Cortafuego, utilizando para ello el módulo de Retroalimentación

- Módulo de Retroalimentación: El Administrador tendrá una consola de mantención del Cortafuego, en el cual podrá eliminar alguna regla que el sistema haya creado en forma automática. Para llevar a cabo esto se utilizará el lenguaje de programación Shell.

## 2.- CARACTERÍSTICAS DEL SISTEMA DE IMPLEMENTACIÓN

Para propósitos de las pruebas que se realizarán se ha implementado una red utilizando máquinas virtuales las características de estas son:

- Virtualización: VMWare Workstation 6.0.5

- Servidores:
    a) Cantidad: 1
    b) Servicios: WEB
    c) Sistema Operativo: Fedora Core 9
    d) Memoria RAM: 256M
    e) Disco Duro: 8GB
    f) Número de Procesadores: 1

- Clientes:
    a) Cantidad: 1
    b) Sistema Operativo: Fedora Core 9

# X. IMPLEMENTACIÓN

**1.- Sensor**

Como se indicó anteriormente, para desarrollar este modulo se utilizaron las herramientas de programación Shell, AWK y utilidades que se encuentran disponibles en un sistema operativo Linux.

La tarea desarrollada por este modulo es: Captura de una trama de tráfico y lanzamiento del modulo de adaptación de la data; lo cual se grafica en el siguiente diagrama:


Figura 26: Diagrama de Flujo Modulo Sensor

El resultado obtenido se encuentra almacenado en el archivo temporal “logeo” y un ejemplo del tráfico capturado se puede observar en el siguiente cuadro:

```
19:26:37.499926 IP 192.168.1.102.45187 > resolv1.terra.cl.domain: 37302+ AAAA? start.fedoraproject.org. (41)
        0x0000:  4500 0045 4fae 4000 4011 5c4e c0a8 0166
        0x0010:  c81c 0481 b083 0035 0031 2a5a 91b6 0100
        0x0020:  0001 0000 0000 0000 0573 7461 7274 0d66
        0x0030:  6564 6f72 6170 726f 6a65 6374 036f 7267
        0x0040:  0000 1c00 01
```

Figura 27: Ejemplo de Data capturada

Una vez capturada esta información, se iniciará el modulo de adaptación de la data.

**2.- Adaptación de la Data**

Este módulo se encuentra dividido en tres partes:

- Almacena la dirección IP origen que envía la data, en un archivo temporal llamado “ip_origen”, el cual será utilizado en la actualización de las reglas del Cortafuego.

- Transforma la información de una base hexadecimal a una base decimal, separando los distintos campos según el protocolo UDP o TCP correspondiente y dándoles el formato requerido por la herramienta de análisis. Esta información es almacenada en un archivo temporal llamado “logeo4”. Sin etiqueta y sin valor en el atributo 0, para que sean la Máquinas de Soporte Vectorial previamente entrenada que decida si corresponde a un tráfico normal o anómalo, según corresponda.

- Posteriormente, se procederá a escalar los valores de cada campo, utilizando para ello el archivo llamado “entrenamiento.scale” con el rango de valores que fue obtenido en la etapa de entrenamiento.

A continuación se presenta un diagrama de flujo con el proceso realizado en el módulo de adaptación de data:

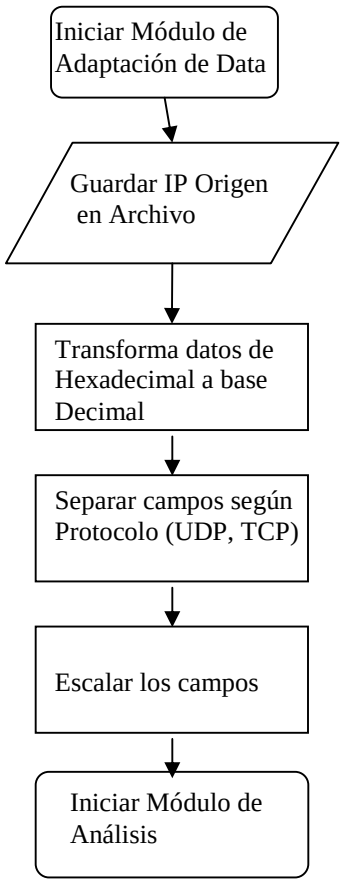


Figura 28: Diagrama de Flujo Módulo Análisis de la Data

A continuación, se presentan algunos ejemplos de capturas:

192.168.1.102

Figura 29: Ejemplo de Dirección IP Origen capturada

1:4 2:5 3:0 4:69 5:20398 6:2 7:0 8:64 9:17 10:23630 11:49320 12:358 13:51228 14:1153 15:0 16:0 17:45187 18:0 19:0 20:0 21:53 22:0

Figura 30: Ejemplo de Data separada en los campos respectivos de los protocolos UDP o TCP

1 4:-0.983333 5:0.646966 6:1 8:-0.487805 9:-1 10:0.609406 11:0.887007 12:-0.99996
13:0.54582 14:-0.853344 15:-1 16:-1 17:-0.799012 18:0.296548 19:-0.428571
20:0.314486 21:-0.972237 22:-0.993873

Figura 31: Ejemplo de data escalada

**3.- Módulo de Análisis**

Una vez obtenida y transformada la data, se procede a clasificarla, a continuación se presenta un diagrama de flujo con la explicación de las funciones que realiza este módulo:

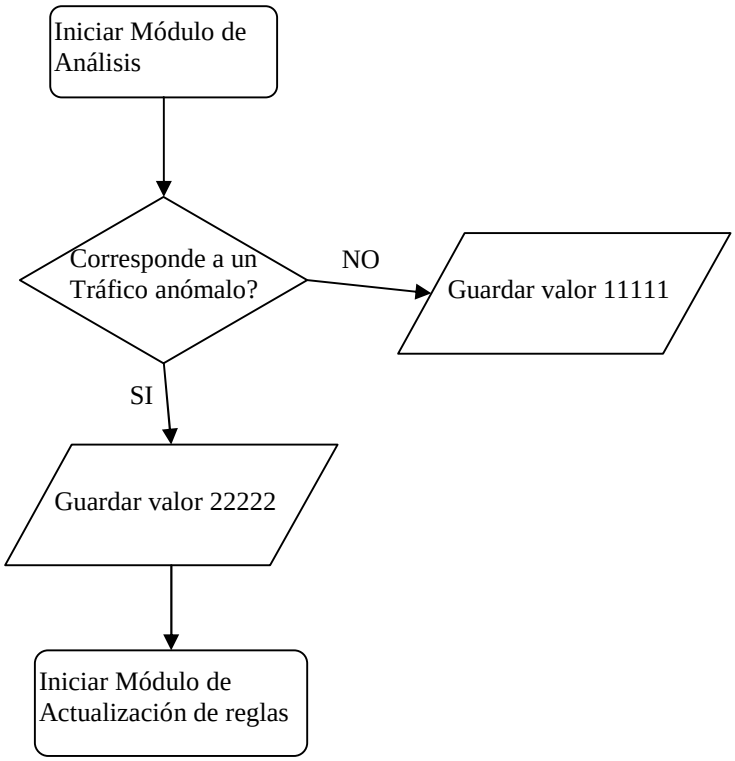


Figura 32: Diagrama de Flujo del módulo de Análisis

Los resultados obtenidos se almacenarán en archivos temporales.

El resultado obtenido para un tráfico normal, se muestra a continuación:

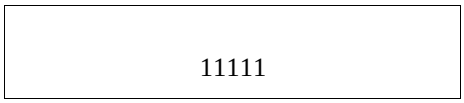


Figura 33: ejemplo del resultado del análisis de un tráfico normal

En el caso de tratarse de un tráfico anómalo, el resultado obtenido será:

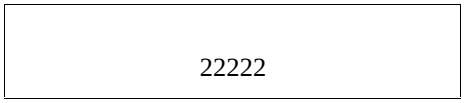


Figura 34: ejemplo del resultado del análisis de un tráfico anómalo

En el caso de tratarse de un tráfico anómalo, es decir donde el archivo temporal contenga un valor igual a “22222”, se procederá a activar el módulo de actualización de reglas,

**4.- Actualización De Reglas**

Una vez activado este módulo, se rescata la dirección IP que se almacenó en el archivo temporal “ip_origen”, y se procede a actualizar las reglas del Cortafuego del servidor, con el siguiente código:

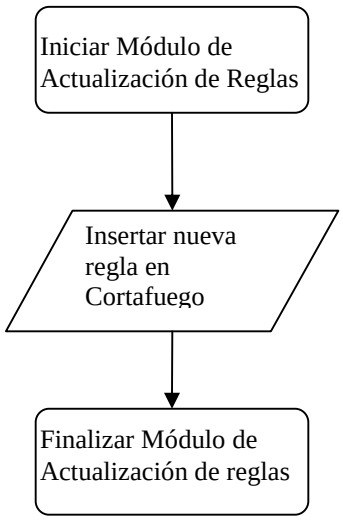


Figura 35: Diagrama de Flujo Módulo de Actualización de Reglas

**5.- Módulo De Retroalimentación**

Este módulo es la herramienta que tendrá el administrador para deshacer alguna decisión que haya tomado el sistema en forma automática, a continuación se muestra un diagrama de flujo del módulo de Retroalimentación:

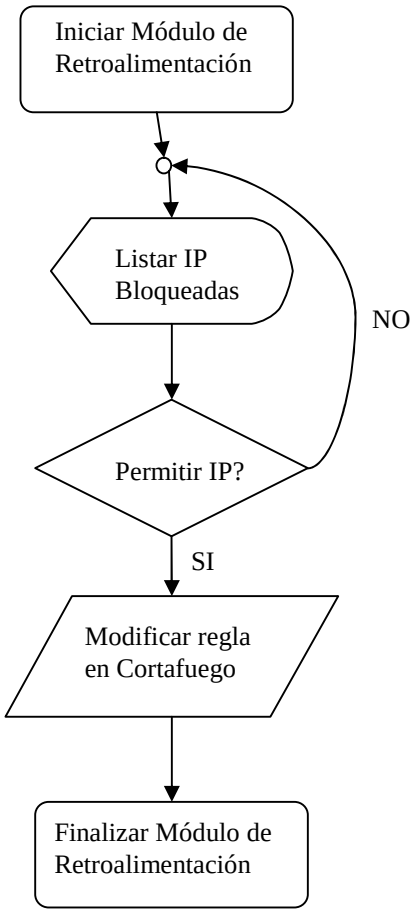


Figura 36: Diagrama de Flujo de Módulo de Administración.

En la figura 29 se muestra la interfaz que tendrá el administrador:

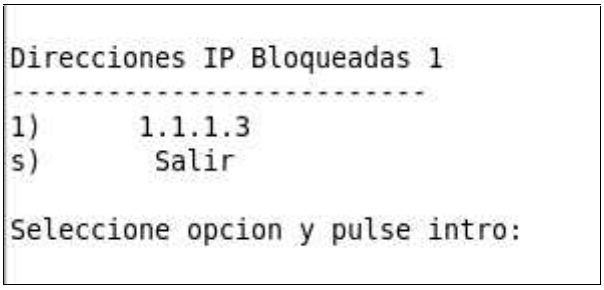


Figura 37: Interfaz de Administración

Y como respuesta a la eliminación de una regla se observará la siguiente pantalla:

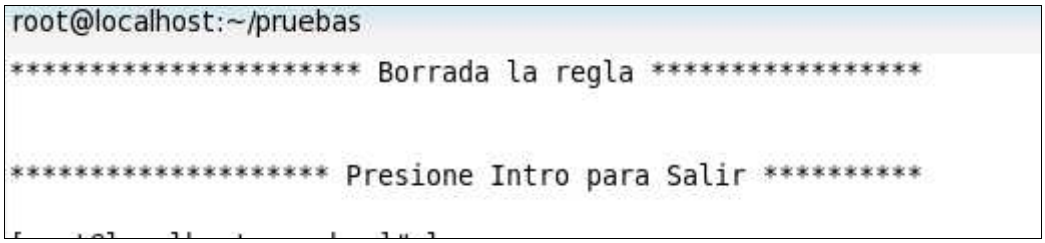


Figura 38: Interfaz de Respuesta a la eliminación de una Regla

**6.- Entrenamiento de las Máquinas de Soporte Vectorial**

Previo al análisis del tráfico, es necesario entrenar a las Máquinas de Soporte Vectorial, por lo cual se implementan los siguientes módulos:

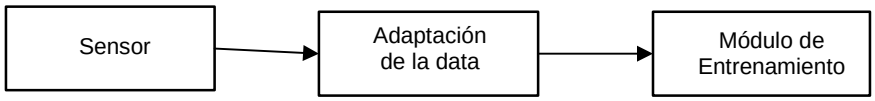


Figura 39: Prototipo de entrenamiento de las Máquinas de Soporte Vectorial

- Sensor: La función que cumple este módulo es igual al módulo de igual nombre que se presentó anteriormente.

  Se procederá a capturar tráfico normal y será almacenado en el archivo "Traf_normal". Posteriormente, se capturará tráfico anómalo y será almacenado en el archivo "Traf_anomalo".

  Una vez capturada esta información, se iniciará el módulo de adaptación de la data.

- Adaptación de la data: La función es similar al módulo del prototipo de análisis de data, con la salvedad que en este módulo, al tratarse de un tráfico anómalo el atributo 0 tendrá un valor definido igual a -1 y su etiqueta poseerá un valor "22222" , por otro lado, en caso de tratarse de un tráfico normal el atributo 0 tendrá un valor de "+1" y su etiqueta respectiva será de "11111".

El resultado obtenido se encuentra almacenado en el archivo temporal “captura” y un ejemplo del tráfico capturado y escalado se puede observar en el siguiente cuadro:

```
11111 0:1 4:-0.830556 5:-0.761748 6:-1 8:-1 9:1 10:0.841696 11:0.887007 12:-1 13:1 14:1
15:-1 16:-1 17:-0.960062 18:-1 19:1 20:-1 21:-0.999997 22:-1
11111 0:1 4:-0.830556 5:-0.761563 6:-1 8:-1 9:1 10:0.84151 11:0.887007 12:-1 13:1 14:1 15:-
1 16:-1 17:-0.960062 18:-1 19:1 20:-1 21:-0.999997 22:-1
11111 0:1 4:-0.830556 5:-0.76144 6:-1 8:-1 9:1 10:0.841386 11:0.887007 12:-1 13:1 14:1 15:-
1 16:-1 17:-0.960062 18:-1 19:1 20:-1 21:-0.999997 22:-1
22222 0:-1 4:-0.983333 5:-0.413429 6:1 8:-0.487805 9:-1 10:-0.496301 11:0.887007 12:-
0.99996 13:-0.808371 14:0.22115 15:-1 16:-0.965109 17:0.123027 18:1 19:-0.428571 20:-1
21:-0.972237 22:-0.999413
22222 0:-1 4:-0.983333 5:0.700028 6:1 8:-0.487805 9:-1 10:-1 11:0.887007 12:-0.99996
13:0.00162096 14:-0.949181 15:-1 16:-0.946574 17:-0.968494 18:-0.897471 19:-0.428571
20:-1 21:-0.972237 22:-0.999377
22222 0:-1 4:-0.781944 5:-1 6:1 8:-0.536585 9:1 10:0.400351 11:0.996388 12:-0.968007
13:0.461796 14:-1 15:-1 16:-1 17:-0.998363 18:-1 19:1 20:-1 21:-0.999903 22:-1
22222 0:-1 4:-0.956944 5:0.851693 6:1 8:-0.487805 9:1 10:0.52543 11:0.887007 12:-0.99996
13:0.54653 14:-0.975573 15:-1 16:-1 17:0.550425 18:-1 19:1 20:-1 21:-1 22:-1
```

Figura 40: Ejemplo de tráfico capturado y etiquetado

Una vez capturada esta información, se reiniciará el sensor hasta que se complete la cantidad definida por el autor como suficiente para el entrenamiento de la Máquinas de Soporte Vectorial, esto es 200 capturas de tráfico normal y 200 capturas de tráfico anómalo. Una vez completada esta cantidad se activará el módulo de entrenamiento.

- Módulo de entrenamiento: Una vez obtenida y transformada la data, se procede a entrenar, propiamente tal, las Máquinas de Soporte Vectorial, este entrenamiento implica crear algunos archivos que se utilizarán posteriormente en el análisis del tráfico con la clasificación del mismo, para esto se utiliza una función Kernel de base radial y con valores Gamma iguales a: 0.001, 0.005, 0.01 y 0.05.

  Se comparará la certeza de la clasificación al utilizar los distintos valores Gamma con el resultado obtenido al utilizar el valor teórico de Gamma, el cual está dado por la formula: 1/k, donde k corresponde a la cantidad de atributos utilizados. En el módulo de adaptación de la data se separan los datos en 22 atributos por lo cual el Gamma teórico a utilizar corresponde a 0,045454.

**Resultados obtenidos en el entrenamiento:**

Primeramente se procede a elegir el valor Gamma a ser utilizado en que se logra un menor nivel de alarmas falsas, ya sean falsos positivos o falsos negativos, el procedimiento utilizado es el siguiente:

- Se capturan 200 paquetes de tráfico normal y 200 paquetes de tráfico anormal. Para un tráfico normal, se realizan consultas WEB desde el PC cliente al servidor con el sistema de entrenamiento en prueba. Por otro lado, para un tráfico anómalo, se realizarán pruebas de PING y FTP.

- Se procede con el modulo de Análisis, y se observa el resultado obtenido. Es decir, se contabilizará la salida obtenida (ya sea un valor de etiqueta igual a 11111 para un tráfico normal y un valor de etiqueta igual a 22222 en el caso de un tráfico anómalo) y que corresponda o no a la data analizada. Se compararán los resultados obtenidos al utilizar los distintos valores de Gamma: 0.001, 0.005, 0.01 y 0.05, antes propuestos.

A continuación se presenta un cuadro resumen con los resultados obtenidos:

| **Prueba Realizada con Tráfico Normal (total de muestras = 200)** | | | | |
|---|---|---|---|---|
| **Clasificación** | **0.01** | **0.05** | **0.1** | **0.5** |
| Anormal | 193 | 8 | 0 | 131 |
| Normal | 7 | 192 | 200 | 69 |
| **% Certeza** | **3.5%** | **96%** | **100%** | **34.5%** |

Tabla 8: Resultados obtenido con un Tráfico Normal y distintos valores Gamma

| Prueba Realizada con Tráfico Anormal (total de muestras = 200) | | | | |
|---|---|---|---|---|
| **Clasificación** | **0.01** | **0.05** | **0.1** | **0.5** |
| Anormal | 0 | 200 | 0 | 0 |
| Normal | 200 | 0 | 200 | 200 |
| **% Certeza** | **0%** | **100%** | **0%** | **0%** |

Tabla 9: Resultados obtenido con un Tráfico Anormal y distintos valores Gamma

Se observa que se logra un mejor desempeño al utilizar un valor de Gamma igual a 0.05. Dicho valor pudo ser calculado en base a que existe una relación inversamente proporcional a la cantidad de atributos utilizados en los vectores, es decir, ya que se utilizan 22 atributos el valor teórico de Gamma sería 0,045454. Por lo cual el valor logrado en forma práctica se aproxima con el valor teórico que se obtendría.

Por lo tanto, el valor de Gamma igual a 0.05 se utilizará en las siguientes pruebas.

**7.- Pruebas Realizadas sobre el Prototipo**

Una vez entrenadas las Máquinas de Soporte Vectorial, se proceden a realizar diversas pruebas del sistema en conjunto propuesto. Se espera que cuando el tráfico se trate de una consulta normal al sistema, el prototipo no reaccione, permita que se complete la conexión y se realicen las consultas respectivas. Por otro lado, si se trata de un tráfico anormal el prototipo debiera reaccionar creando las reglas restrictivas en el cortafuego que impida que la conexión se complete, evitando que el ataque siga su curso, y con ello un daño mayor.

Las pruebas se han realizando aumentando la complejidad del tráfico a detectar. A continuación se presentan las pruebas realizadas:

a) Se realizan consultas HTTP desde el computador cliente al servidor, el prototipo permite que se complete la conexión y se reciban las consultas.
b) Se realiza un envío de mensajes utilizando el protocolo ICMP (PING), se observa que al tercer mensaje se ha bloqueado la comunicación, es decir se ha creado la regla en el cortafuego que bloquea la conexión desde el computador cliente al servidor para todo protocolo.
c) Se procede a escanear los puertos desde el computador cliente hacia el servidor, utilizando la herramienta NMAP, no se obtiene respuesta del servidor pero se observa que se ha creado la regla respectiva en el cortafuego.

Las siguientes pruebas se realizan utilizando la aplicación Backtrack 3, la cual corresponde a un conjunto de herramientas de análisis de seguridad y monitoreo de vulnerabilidades. Las herramientas utilizadas para realizar intentos de intrusiones al servidor de prototipo, son:

- XPROBE2: Permite escanear una red remota y entrega información sobre el Sistema Operativo utilizado.
- METASPLOIT: Es una herramienta Open Source de seguridad informática que proporciona información acerca de vulnerabilidades de seguridad. Es útil en pruebas de penetración y en el desarrollo de firmas para Sistemas de Detección de Intrusos.
- PHP Search Module SQL Injection: Esta herramienta busca ingresar múltiples comandos SQL a través de una vulnerabilidad que se encuentra en ciertas versiones de PHP.
- PHP viewtopic.php Arbitry code Execution: Esta herramienta utiliza la falla de seguridad en que el parámetro highlight en el script viewtopic.php, el cual no es analizado correctamente y se podría inyectar algún comando aleatorio.

Los resultados obtenidos fueron:

d) XPROBE2: El prototipo a pesar de que entrega las respuestas de los puertos que se encuentran en abiertos en el servidor de pruebas, alcanza a bloquear la dirección IP.

e) METASPLOIT: En este caso el prototipo alcanza a bloquear la dirección IP del atacante antes que éste último comience a ejecutar los script de las vulnerabilidades encontradas.

f) PHP Search Module SQL Injection y PHP viewtopic.php Arbitry code Execution: En ambos casos el prototipo no detecta el ataque por lo cual se lleva a cabo el proceso de intrusión.

# XI. ANÁLISIS DE LOS RESULTADOS

La implementación del prototipo se realizó en un ambiente de CLI del sistema operativo Linux Fedora Core9, por lo cual dicho prototipo no se presenta un ambiente gráfico ni herramientas visuales como para desarrollar gráficos y análisis posteriores.

El análisis desarrollado con el prototipo se basó en conocimiento empírico obtenido de las distintas pruebas que se desarrollaron. Estas pruebas abarcaron la comprobación del correcto funcionamiento al escoger el valor de Gamma en la función del Kernel a utilizar, resultados que se entregaron en las tablas 8 y 9, hasta el realizar pruebas prácticas de envío de consultas normales y de algunos tipos de ataques desde un computador cliente al servidor con el prototipo alojado, verificando la correcta clasificación y posterior creación de la regla en el cortafuego.

Por lo anterior, ha sido necesario recrear dicho prototipo en otras herramientas gráficas y en un ambiente Windows. Por lo cual, se han utilizado aplicaciones como WEKA y Excel, para la elaboración de esta sección.

Los distintos atributos utilizados del encabezamiento de los protocolos TCP, UDP, IP se han enumerado, por un tema de simplicidad de la anotación, por lo cual, en las siguientes tablas y gráficos se hablará de campo 1 al campo 22. La tabla 10 relaciona los campos del 1 al 22 con los atributos del encabezamiento de los protocolos analizados:

| | | |
|---|---|---|
| Campo 1 | Versión | Protocolo IP |
| Campo 2 | IHL | Protocolo IP |
| Campo 3 | Tipo de Servicio | Protocolo IP |
| Campo 4 | Largo Total | Protocolo IP |
| Campo 5 | Identificación | Protocolo IP |
| Campo 6 | Bandera | Protocolo IP |
| Campo 7 | Fragmento de Compensación | Protocolo IP |
| Campo 8 | Tiempo de Vida | Protocolo IP |
| Campo 9 | Protocolo | Protocolo IP |
| Campo 10 | Chequeo de encabezamiento | Protocolo IP |
| Campo 11 | Número de Puerto del Origen | Protocolo TCP y UDP |
| Campo 12 | Número de Puerto del Destino | Protocolo TCP y UDP |
| Campo 13 | Número de Secuencia | Protocolo TCP |
| Campo 14 | Número de Reconocimiento | Protocolo TCP |
| Campo 15 | Offset | Protocolo TCP |
| Campo 16 | Bits de Control | Protocolo TCP |
| Campo 17 | Bits de Control | Protocolo TCP |
| Campo 18 | Bits de Control | Protocolo TCP |
| Campo 19 | Bits de Control | Protocolo TCP |
| Campo 20 | Window | Protocolo TCP |
| Campo 21 | Checksum | Protocolo TCP y UDP |
| Campo 22 | Puntero Urgente | |

Tabla 10: Relación de campos con atributos.

**1.- Naturaleza de la Data**

Desde la tabla 11 a la tabla 13 se presenta un extracto de los datos utilizados para el entrenamiento de las Máquinas de Soporte Vectorial.

| Campo1 | Campo2 | Campo3 | Campo4 | Campo5 | Campo6 | Campo7 | Campo8 | Etiqueta |
|---|---|---|---|---|---|---|---|---|
| 4 | 5 | 0 | 52 | 19036 | 2 | 0 | 64 | NOR |
| 4 | 5 | 0 | 52 | 55171 | 2 | 0 | 64 | NOR |
| 4 | 5 | 0 | 197 | 0 | 2 | 0 | 58 | NOR |
| 4 | 5 | 0 | 71 | 60093 | 2 | 0 | 64 | NOR |
| 4 | 5 | 0 | 52 | 8514 | 2 | 0 | 64 | NOR |
| 4 | 5 | 0 | 162 | 3836 | 0 | 0 | 1 | NOR |
| 4 | 5 | 0 | 52 | 5880 | 2 | 0 | 48 | NOR |
| 4 | 5 | 0 | 68 | 32486 | 2 | 0 | 64 | NOR |
| 4 | 5 | 0 | 1480 | 50432 | 2 | 0 | 47 | NOR |
| 4 | 5 | 0 | 1480 | 22652 | 2 | 0 | 47 | NOR |
| 4 | 5 | 0 | 197 | 0 | 2 | 0 | 58 | NOR |
| 4 | 5 | 0 | 614 | 50442 | 2 | 0 | 47 | NOR |
| 4 | 5 | 0 | 52 | 44983 | 2 | 0 | 64 | NOR |
| 4 | 5 | 0 | 52 | 44480 | 2 | 0 | 64 | NOR |
| 4 | 5 | 0 | 1480 | 3648 | 2 | 0 | 47 | NOR |
| 4 | 5 | 0 | 52 | 12705 | 2 | 0 | 64 | NOR |
| 4 | 5 | 0 | 52 | 37537 | 2 | 0 | 64 | NOR |
| 4 | 5 | 0 | 52 | 12738 | 2 | 0 | 64 | NOR |
| 4 | 5 | 0 | 52 | 12739 | 2 | 0 | 64 | NOR |
| 4 | 5 | 0 | 52 | 60005 | 2 | 0 | 64 | NOR |
| 4 | 5 | 0 | 52 | 0 | 2 | 0 | 47 | NOR |
| 4 | 5 | 0 | 52 | 12706 | 2 | 0 | 64 | NOR |
| 4 | 5 | 0 | 52 | 12741 | 2 | 0 | 64 | NOR |
| 4 | 5 | 0 | 52 | 60007 | 2 | 0 | 64 | NOR |
| 4 | 5 | 0 | 52 | 44483 | 2 | 0 | 64 | NOR |
| 4 | 5 | 0 | 52 | 25988 | 2 | 0 | 64 | NOR |
| 4 | 5 | 0 | 52 | 37540 | 2 | 0 | 64 | NOR |
| 4 | 5 | 0 | 511 | 38970 | 2 | 0 | 64 | NOR |
| 4 | 5 | 0 | 52 | 2588 | 2 | 0 | 64 | NOR |

Tabla 11: Ejemplo de los Datos utilizados en el entrenamiento de las Máquinas de Soporte Vectorial (Campos 1 al 8), con su respectiva etiqueta

| Campo9 | Campo10 | Campo11 | Campo12 | Campo13 | Campo14 | Campo15 | Campo16 | Etiqueta |
|---|---|---|---|---|---|---|---|---|
| 6 | 16533 | 49320 | 357 | 20719 | 40150 | 2.71E+14 | 36690841 | NOR |
| 6 | 296 | 49320 | 357 | 38958 | 2013 | 2.87E+14 | 56181663 | NOR |
| 17 | 45437 | 51228 | 1153 | 49320 | 357 | 0 | 0 | NOR |
| 17 | 49469 | 49320 | 357 | 51228 | 1153 | 0 | 0 | NOR |
| 6 | 57252 | 49320 | 357 | 19069 | 11603 | 2.82E+14 | 2.43E+14 | NOR |
| 17 | 63560 | 49320 | 356 | 61439 | 65530 | 0 | 0 | NOR |
| 6 | 33785 | 20719 | 40150 | 49320 | 357 | 5299260 | 2.86E+13 | NOR |
| 17 | 11543 | 49320 | 357 | 51228 | 1154 | 0 | 0 | NOR |
| 6 | 7959 | 38958 | 2013 | 49320 | 357 | 5302430 | 551998056 | NOR |
| 6 | 35739 | 38958 | 2013 | 49320 | 357 | 5302431 | 550259030 | NOR |
| 17 | 45437 | 51228 | 1153 | 49320 | 357 | 0 | 0 | NOR |
| 6 | 8815 | 38958 | 2013 | 49320 | 357 | 5302430 | 552007850 | NOR |
| 6 | 56121 | 49320 | 357 | 20719 | 40150 | 3.69E+14 | 2.50E+14 | NOR |
| 6 | 56624 | 49320 | 357 | 20719 | 40150 | 3.70E+14 | 2.51E+14 | NOR |
| 6 | 54743 | 38958 | 2013 | 49320 | 357 | 5292852 | 583214884 | NOR |
| 6 | 50768 | 49320 | 357 | 16931 | 16034 | 2.68E+13 | 2.48E+14 | NOR |
| 6 | 17930 | 49320 | 357 | 38958 | 2013 | 3.90E+14 | 2.90E+14 | NOR |
| 6 | 52972 | 49320 | 357 | 19069 | 11659 | 2.73E+14 | 2.99E+14 | NOR |
| 6 | 52971 | 49320 | 357 | 19069 | 11659 | 2.73E+14 | 2.99E+14 | NOR |
| 6 | 60997 | 49320 | 357 | 38958 | 2013 | 3.28E+14 | 3.38E+14 | NOR |
| 6 | 59819 | 38958 | 2013 | 49320 | 357 | 5302435 | 545516257 | NOR |
| 6 | 50767 | 49320 | 357 | 16931 | 16034 | 2.68E+13 | 2.48E+14 | NOR |
| 6 | 52969 | 49320 | 357 | 19069 | 11659 | 2.73E+14 | 2.99E+14 | NOR |
| 6 | 60995 | 49320 | 357 | 38958 | 2013 | 3.28E+14 | 3.38E+14 | NOR |
| 6 | 56621 | 49320 | 357 | 20719 | 40150 | 3.70E+14 | 2.51E+14 | NOR |
| 6 | 37485 | 49320 | 357 | 16931 | 16034 | 3.88E+14 | 837211092 | NOR |
| 6 | 17927 | 49320 | 357 | 38958 | 2013 | 3.90E+14 | 2.90E+14 | NOR |
| 6 | 16038 | 49320 | 357 | 38958 | 2013 | 3.28E+14 | 684070874 | NOR |
| 6 | 60885 | 49320 | 357 | 16931 | 16034 | 3.88E+14 | 837862058 | NOR |

Tabla 12: Ejemplo de los Datos utilizados en el entrenamiento de las Máquinas de Soporte Vectorial (Campos 9 al 16), con su respectiva etiqueta

| Campo17 | Campo18 | Campo19 | Campo20 | Campo21 | Campo22 | Etiqueta |
|---|---|---|---|---|---|---|
| 36358 | 16616576 | -5.48E+16 | 2.56E+14 | 16844810 | 582835 | NOR |
| 1020 | 851840 | -5.48E+16 | 3.79E+14 | 16844810 | 618650 | NOR |
| 53 | 0 | 0 | 0 | 58765 | 0 | NOR |
| 50195 | 0 | 0 | 0 | 53 | 0 | NOR |
| 1140 | 15243136 | -5.48E+16 | 3.30E+14 | 16844810 | 723428 | NOR |
| 1235 | 0 | 0 | 0 | 1900 | 0 | NOR |
| 38087 | 13113984 | -5.48E+16 | 2.62E+14 | 16844810 | 890006238 | NOR |
| 44465 | 0 | 0 | 0 | 53 | 0 | NOR |
| 43456 | 14224512 | -5.48E+16 | 493748224 | 16844810 | 1148391127 | NOR |
| 44333 | 15318400 | -5.48E+16 | 2.35E+14 | 16844810 | 1148391943 | NOR |
| 53 | 0 | 0 | 0 | 38403 | 0 | NOR |
| 43456 | 14676096 | -5.48E+16 | 1363279872 | 16844810 | 1148393153 | NOR |
| 43561 | 10780032 | -5.48E+16 | 580124672 | 16844810 | 904571 | NOR |
| 43580 | 2841984 | -5.48E+16 | 1657602048 | 16844810 | 907932 | NOR |
| 51154 | 12533632 | -5.48E+16 | 3.64E+14 | 16844810 | 1148398046 | NOR |
| 32461 | 13323392 | -5.48E+16 | 769130496 | 16844810 | 923298 | NOR |
| 8449 | 7790720 | -5.48E+16 | 115212288 | 16844810 | 924795 | NOR |
| 49838 | 2749312 | -5.48E+16 | 2.63E+14 | 16844810 | 927102 | NOR |
| 49838 | 2749312 | -5.48E+16 | 2.24E+14 | 16844810 | 933050 | NOR |
| 8903 | 8977792 | -5.48E+16 | 4.15E+14 | 16844810 | 934562 | NOR |
| 44360 | 5004928 | -5.48E+16 | 2.29E+14 | 16844810 | 1148404821 | NOR |
| 32461 | 13323392 | -5.48E+16 | 2.83E+14 | 16844810 | 957410 | NOR |
| 49838 | 2749312 | -5.48E+16 | 4.19E+14 | 16844810 | 968739 | NOR |
| 8903 | 8977792 | -5.48E+16 | 3.76E+13 | 16844810 | 1006096 | NOR |
| 43580 | 2841984 | -5.48E+16 | 1261633536 | 16844810 | 1045044 | NOR |
| 35142 | 4290688 | -5.48E+16 | 3.36E+14 | 16844810 | 1048681 | NOR |
| 8449 | 7790720 | -5.48E+16 | 179699712 | 16844810 | 1054881 | NOR |
| 10459 | 5710464 | -5.48E+16 | 2.32E+14 | 16844810 | 1057318 | NOR |
| 35224 | 2881408 | -5.48E+16 | 2.95E+14 | 16844810 | 1059923 | NOR |

Tabla 13: Ejemplo de los Datos utilizados en el entrenamiento de las Máquinas de Soporte Vectorial (Campos 17 al 22), con su respectiva etiqueta

Utilizando la aplicación WEKA se ha logrado representar cada campo, y su relación con la clasificación final. Tanto en los ejes X e Y se grafican los valores del campo analizado, en colores rojos y azules se indican las clases a la que pertenecen dichos valores, ya sea normal o anormal respectivamente.

En las figuras 41 a la 56 se presenta cada campo.

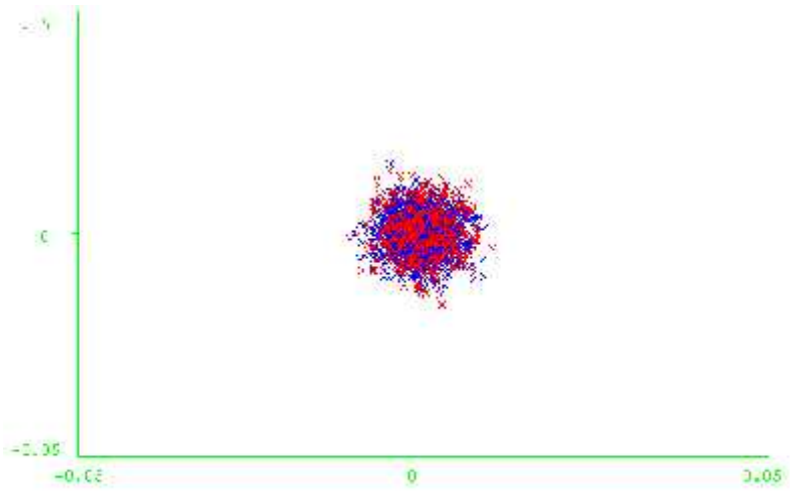

Figura 41 Representación Campos 1, 2, 3, 4, 8

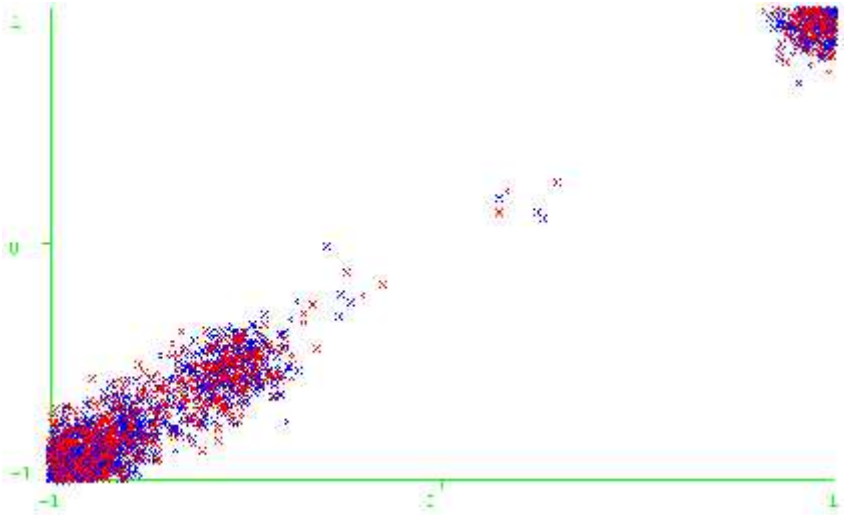

Figura 42: Representación Campo 5

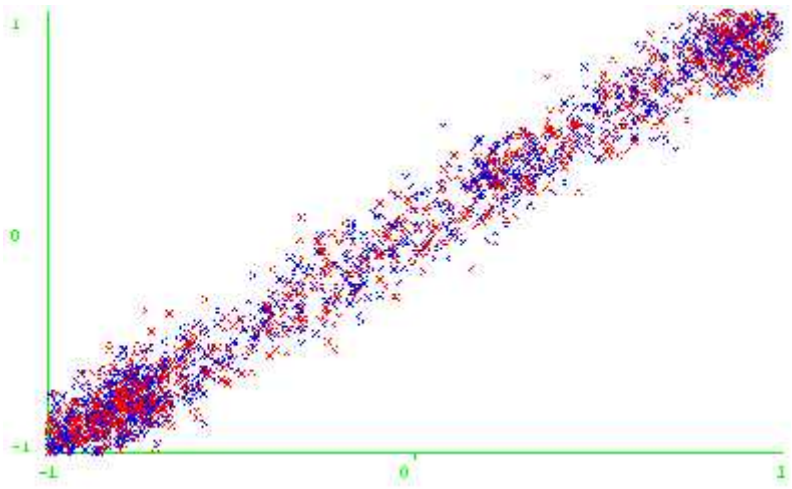


Figura 43: Representación Campo 6, 11

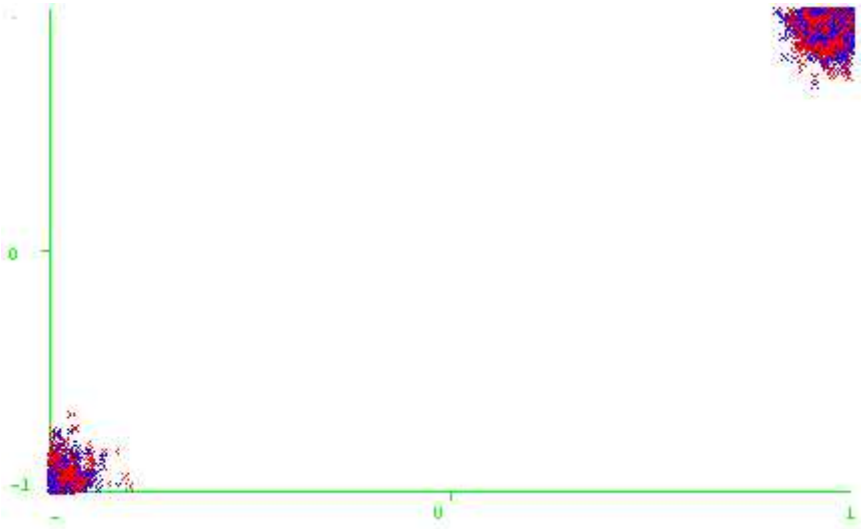


Figura 44: Representación Campo 7, 10

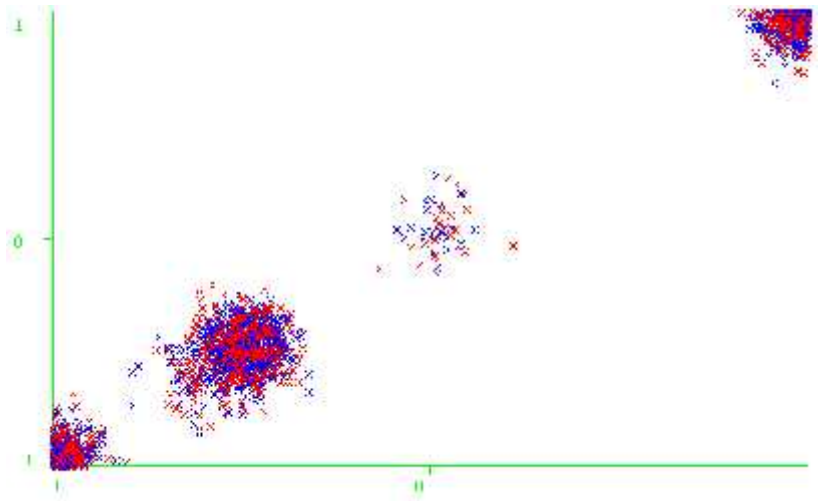

Figura 45: Representación Campo 9

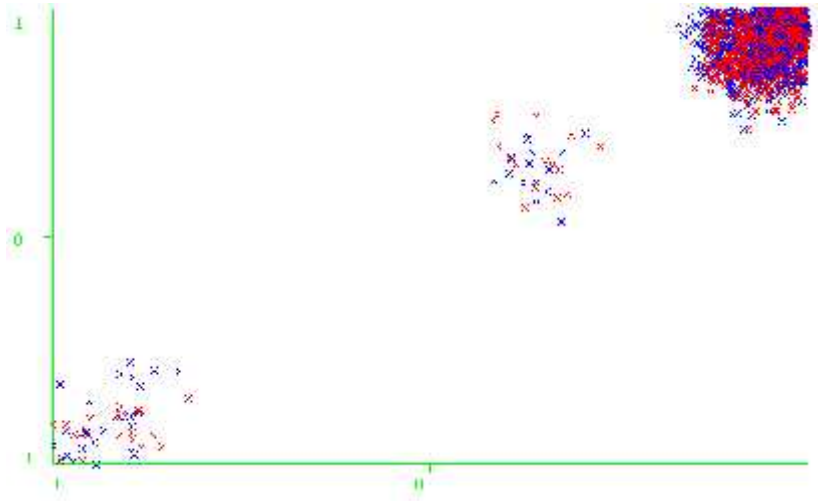

Figura 46: Representación Campo 12

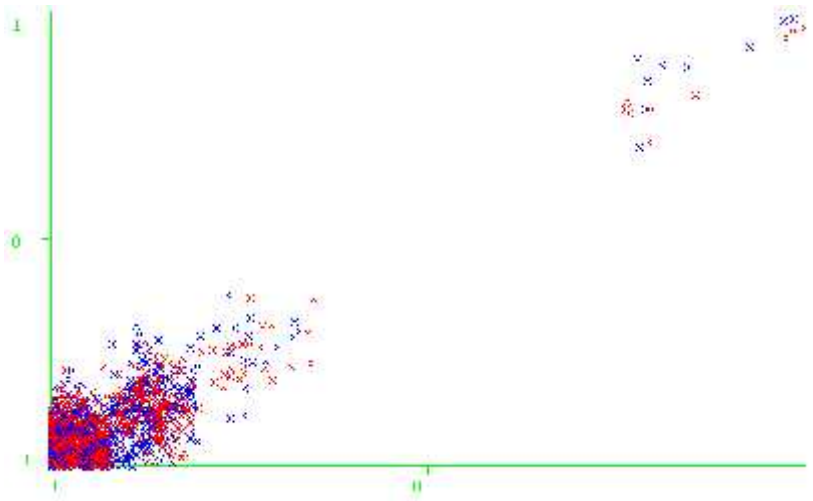

Figura 47: Representación Campo 13

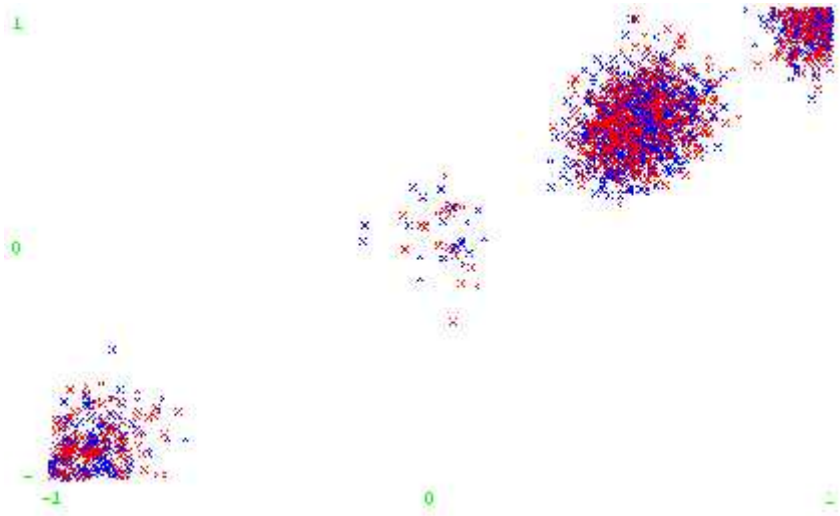

Figura 48: Representación Campo 14

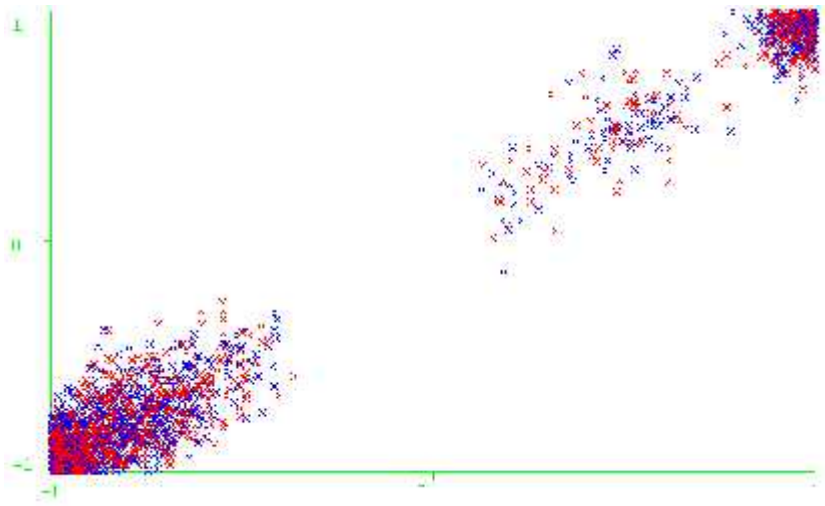

Figura 49: Representación Campo 15

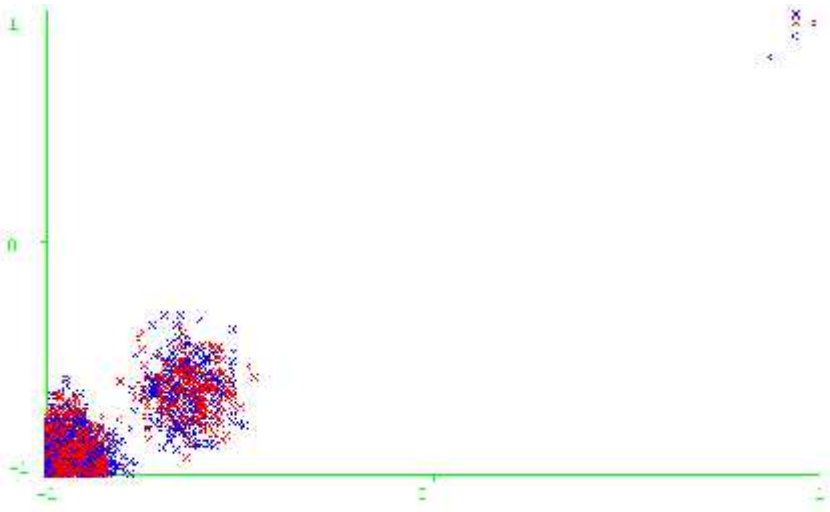

Figura 50: Representación Campo 16

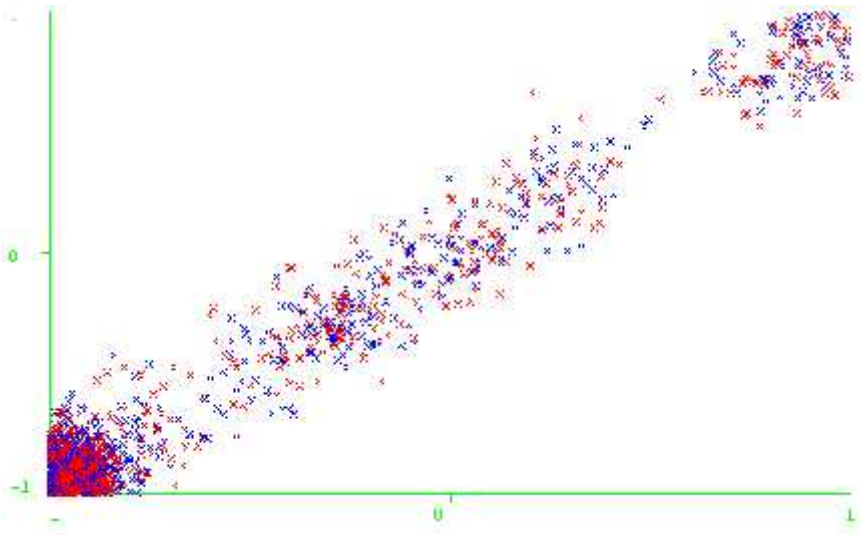


Figura 51: Representación Campo 17

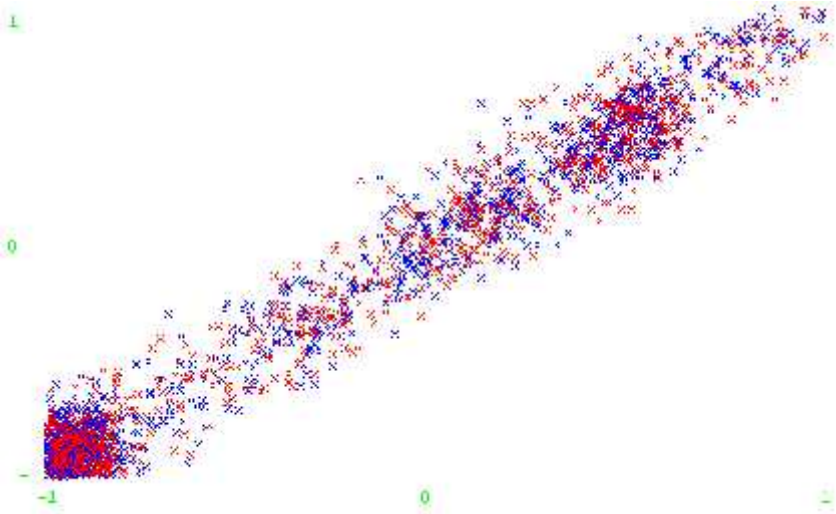


Figura 52: Representación Campo 18

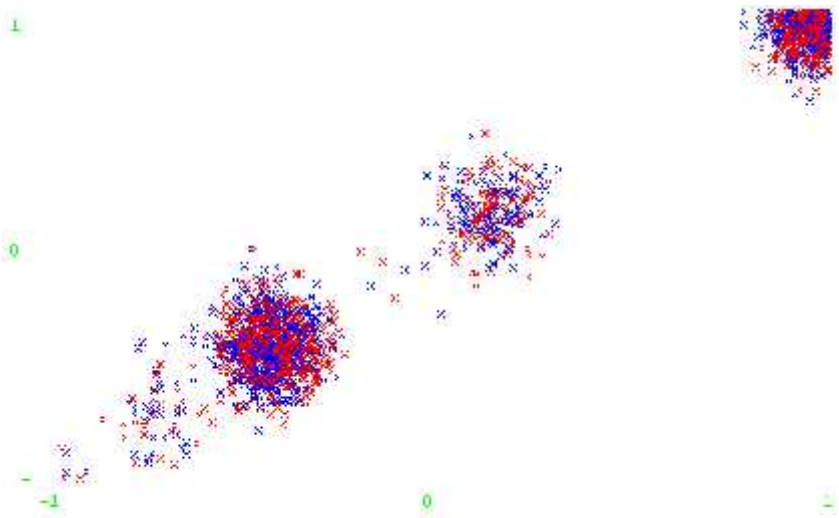


Figura 53: Representación Campo 19

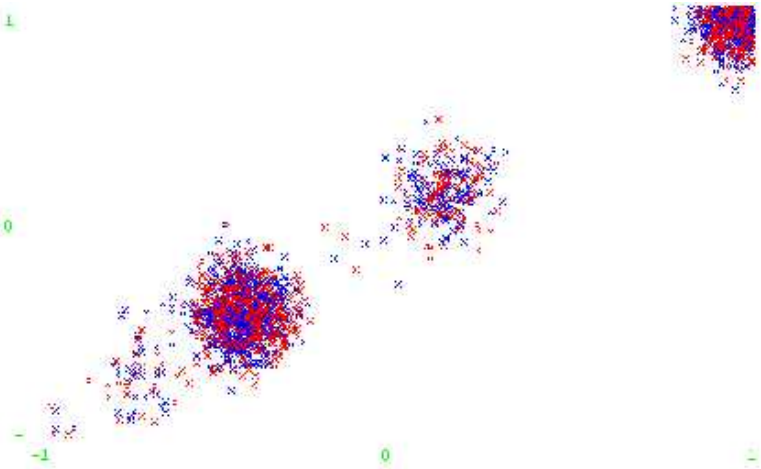


Figura 54: Representación Campo 20

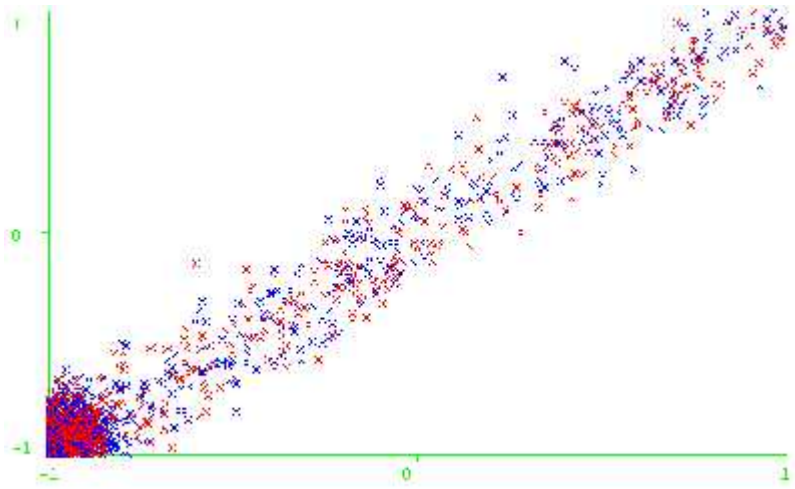

Figura 55: Representación Campo 21

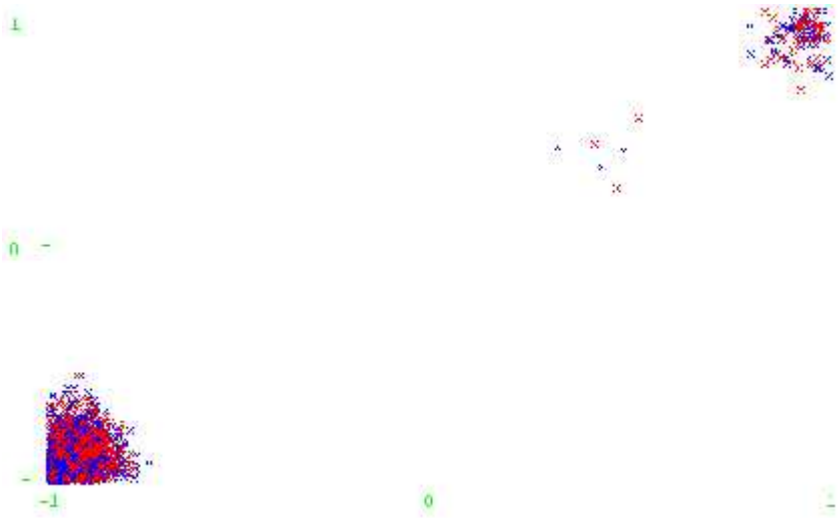

Figura 56: Representación Campo 22

Se puede observar que en las distintas dimensiones de los atributos, la data no puede ser separada o clasificada con alguna función lineal.

Por lo que se demuestra que se requiere la utilización de funciones Kernel, en la clasificación con Máquinas de Soporte Vectorial.

**2.- Incidencia del atributo en la clasificación**

A continuación, se procede a graficar las curvas de ROC (Receiver Operating Characteristic) y su área bajo la curva (AUC). Para ello se utilizará Microsoft Excel. Y el área será calculada usada una integración trapezoidal, a través de la ecuación 39:

$$AUC = (TPR(i) + TPR(i+1)/2)*(FPR(i)-FPR(i+1) \qquad \text{(Ec 39)}$$

Donde,

TPR: Tasa de Verdaderos Positivos

FPR: Tasa de Falsos Positivos

Los valores se presentan en el anexo A.

En la tabla 12 se presenta un resumen de las áreas bajo la curva obtenidas para cada campo.

| Campo | AUC |
|---|---|
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0.42898935 |
| 5 | 0.47268639 |
| 6 | 0 |
| 7 | 0 |
| 8 | 0.32434083 |
| 9 | 0 |
| 10 | 0.49692308 |
| 11 | 0.49339172 |
| 12 | 0.00144852 |
| 13 | 0.02201183 |
| 14 | 0.02201183 |
| 15 | 0.23750059 |
| 16 | 0.00146272 |
| 17 | 0.46969941 |
| 18 | 0.23751479 |
| 19 | 0.02201183 |
| 20 | 0.23751479 |
| 21 | 0.00145799 |
| 22 | 0.00146272 |

Tabla 12: AUC para cada campo.

Una vez calculados los valores de TPR y FPR, eliminarán los valores repetidos, el resultado de esta acción se etiqueta como TPRu y FPRu. Con estos datos se ha procedido a graficar las curvas ROC, donde el eje Y se

representan los valores de TPRu y en los ejes X se representan los valores de FPRu, las figuras 57 a la 72 dan cuenta de esto:

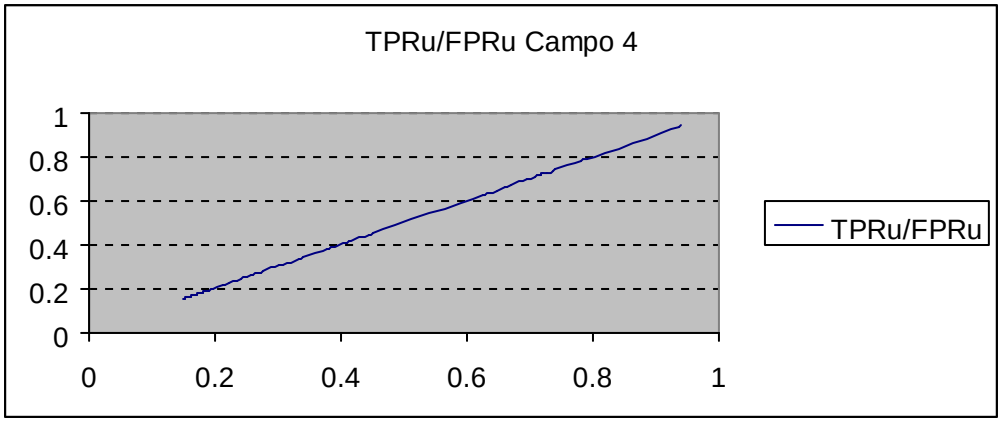


Figura 57: Curva ROC con Campo 4

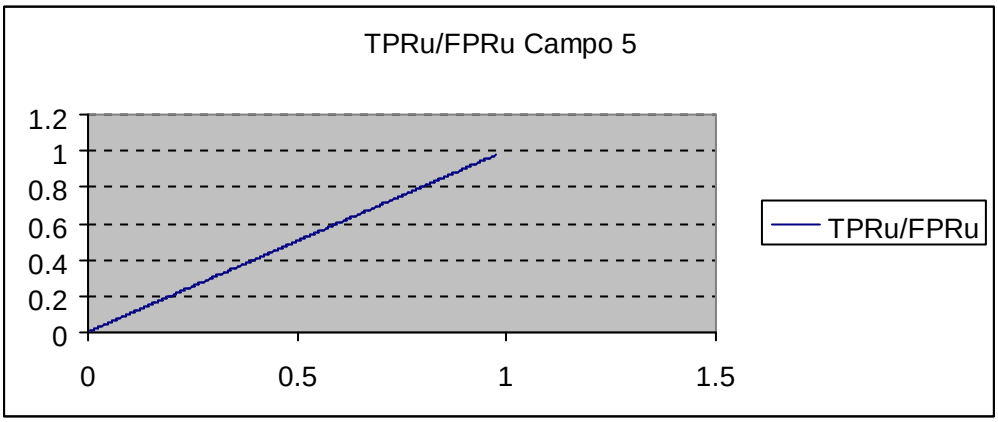


Figura 58: Curva ROC con Campo 5

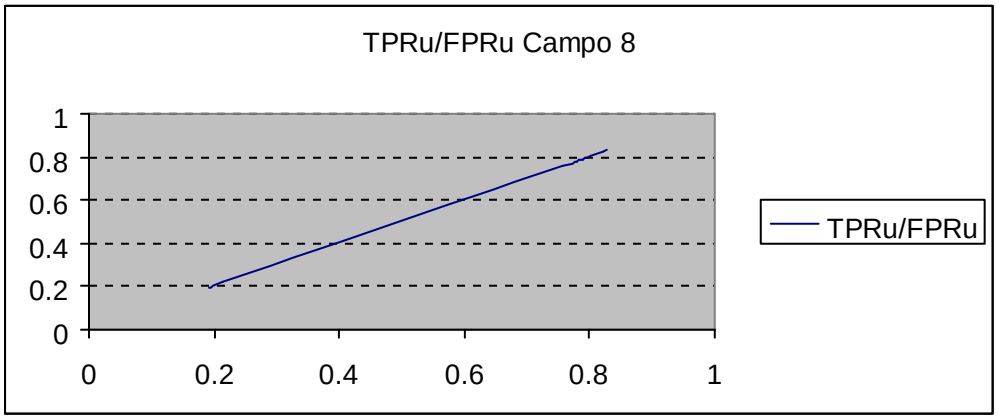


Figura 59: Curva ROC con Campo 8

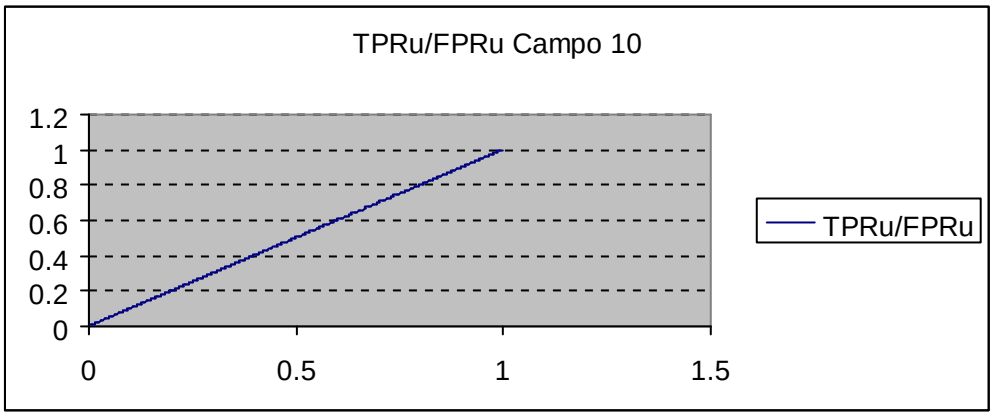


Figura 60: Curva ROC con Campo 10

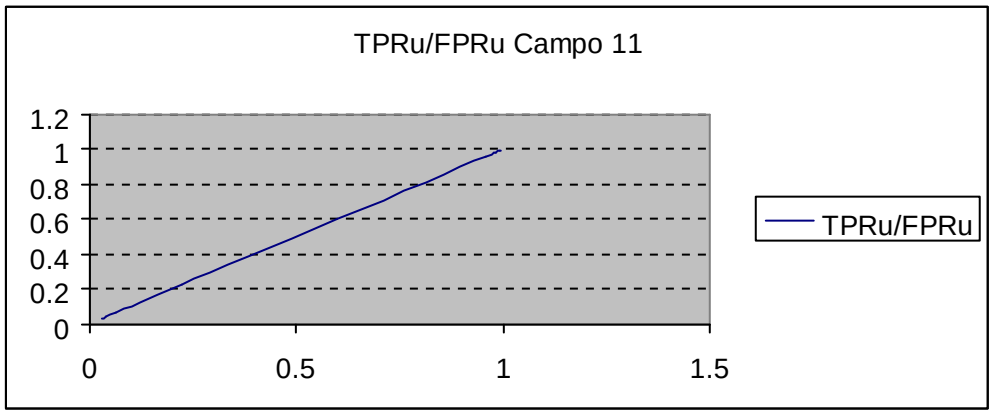


Figura 61: Curva ROC con Campo 11

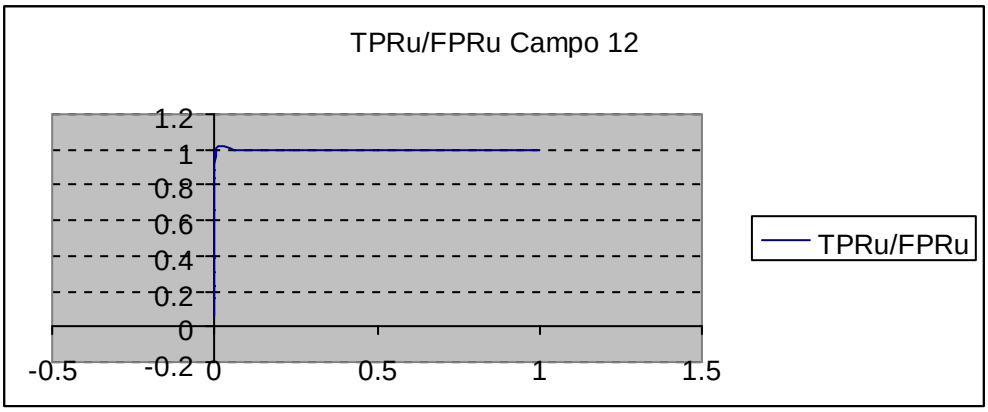


Figura 62: Curva ROC con Campo 12

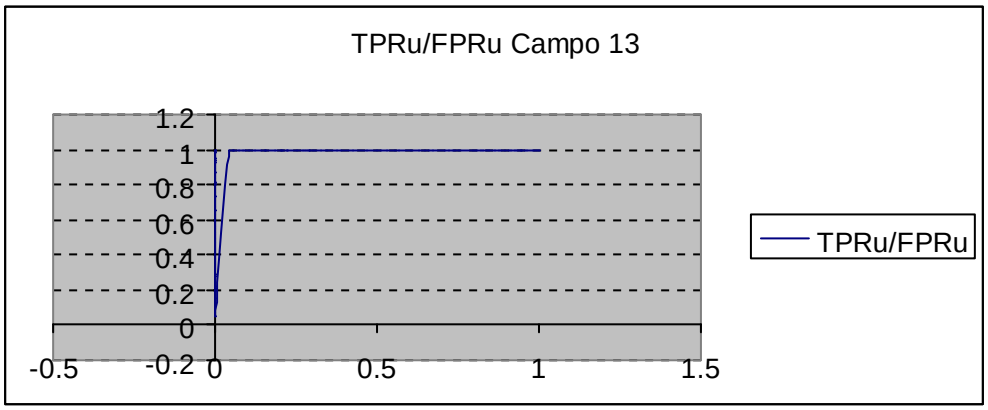


Figura 63: Curva ROC con Campo 13

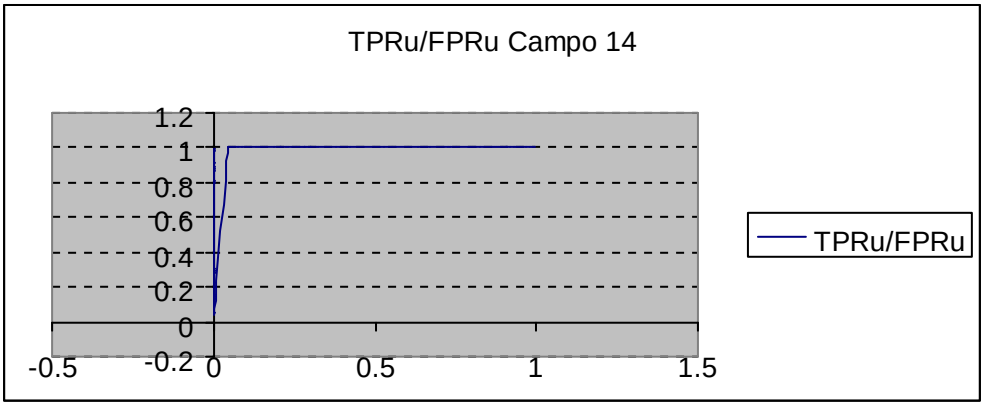


Figura 64: Curva ROC con Campo 14

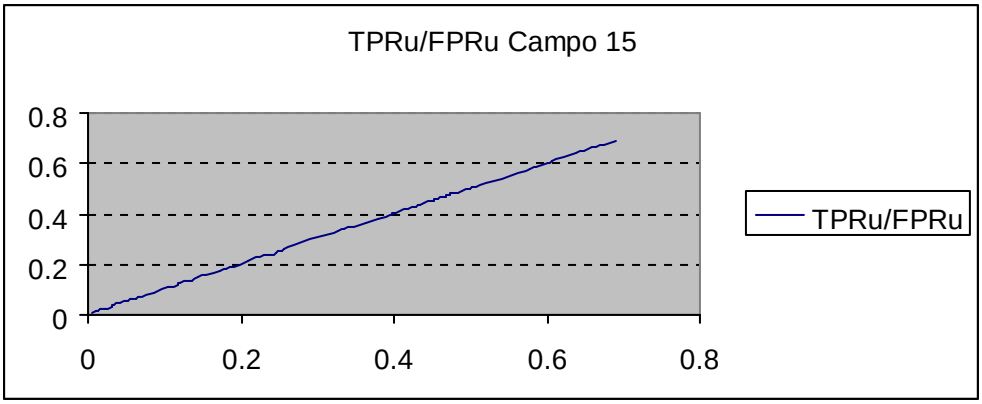


Figura 65: Curva ROC con Campo 15

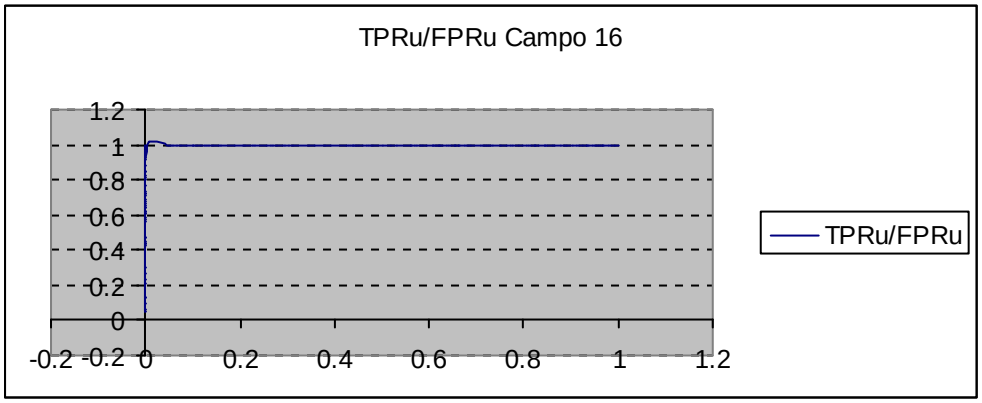


Figura 66: Curva ROC con Campo 16

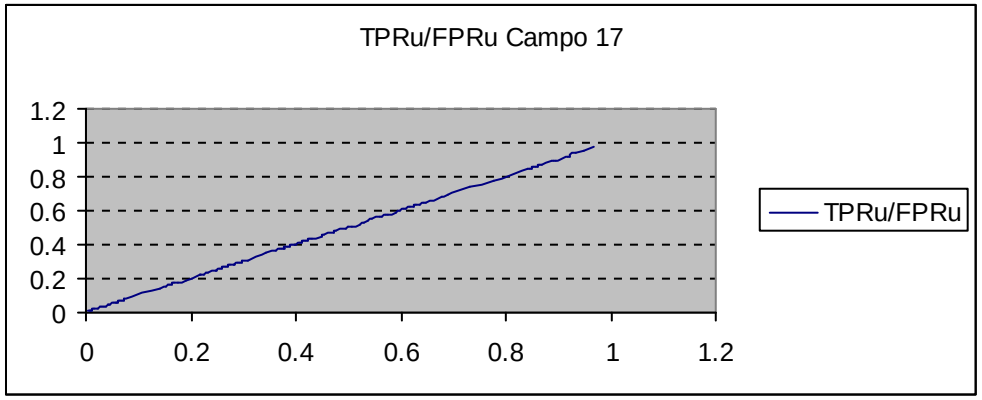


Figura 67: Curva ROC con Campo 17

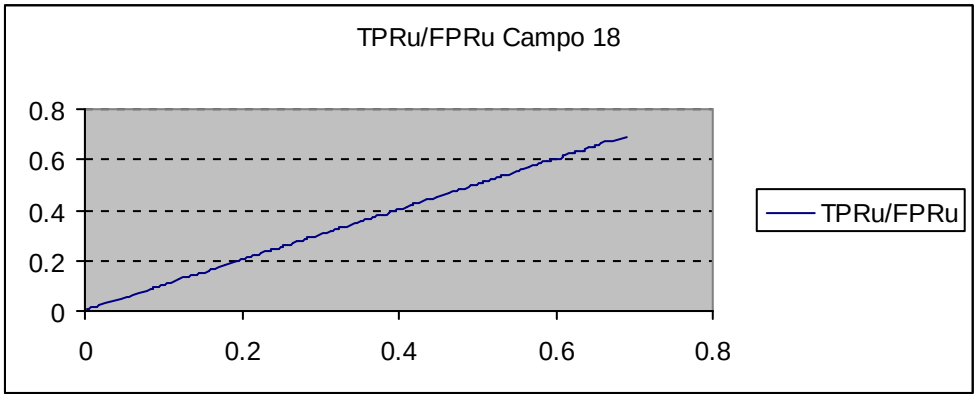


Figura 68: Curva ROC con Campo 18

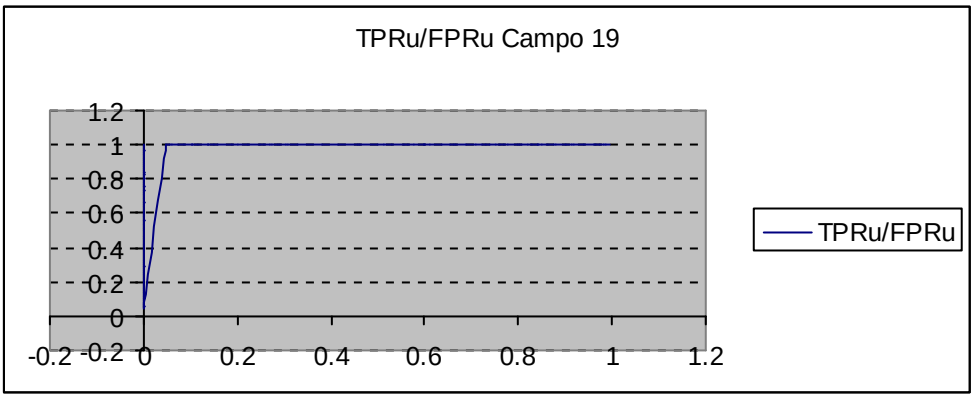


Figura 69: Curva ROC con Campo 19

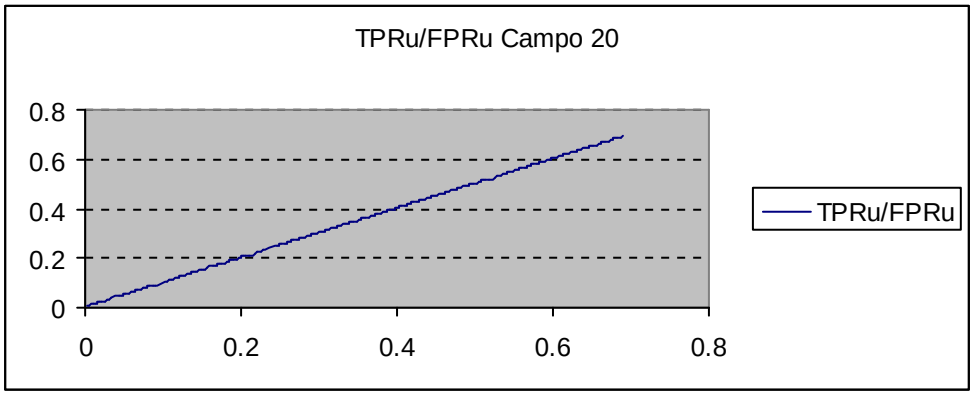


Figura 70: Curva ROC con Campo 20

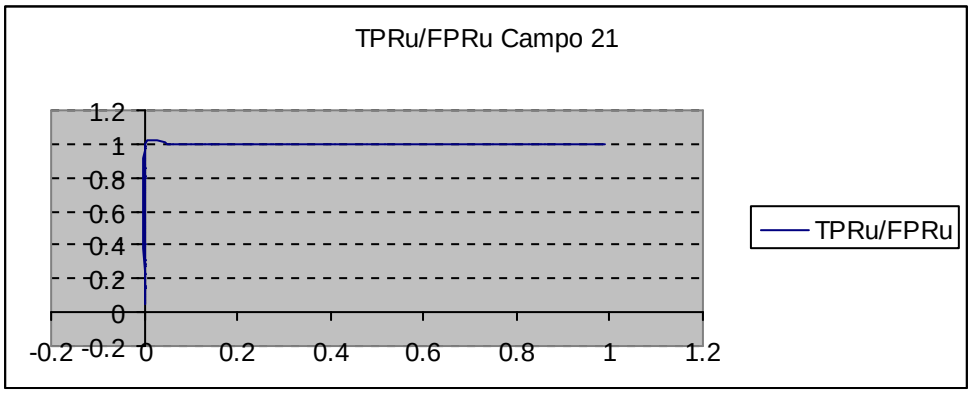


Figura 71: Curva ROC con Campo 21

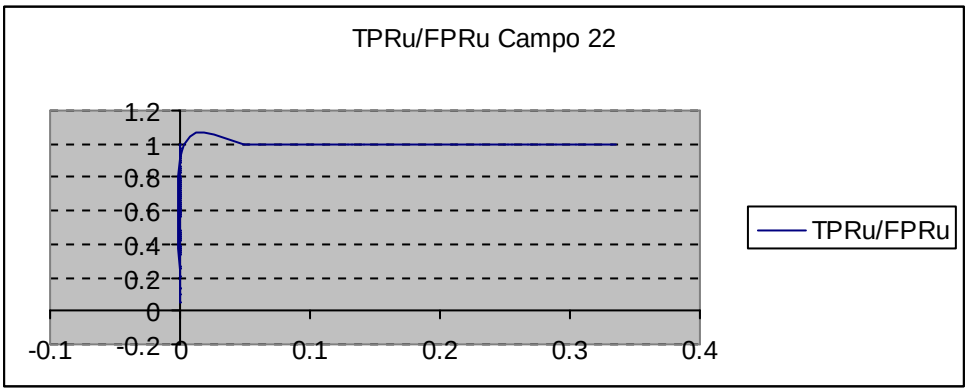


Figura 72: Curva ROC con Campo 22

De los gráficos obtenidos se puede inferir que los atributos de los campos 12, 13, 14, 16, 19, 21 y 22; poseen un altísima incidencia en la correcta clasificación de la data, esto es observando que el área bajo la curva es el mayor posible. Por lo cual deben ser siempre considerados en la construcción de los vectores.

Por otro lado, se puede observar que los atributos de los campos 1, 2, 3, 6, 7 y 9; no tienen incidencia alguna en la correcta clasificación, esto se puede aseverar debido a que el área bajo la curva es igual a cero. Por lo cual perfectamente se pueden omitir dichos parámetros y el resultado no variaría.

**3.- Curva ROC del Modelo**

El procedimiento realizado con cada campo será replicado utilizando el modelo del prototipo en conjunto, para lo cual se utiliza el Programa WEKA.

A continuación, se presenta la curva obtenida con dicho programa. El eje Y representa TPR y el eje X corresponde a los valores FPR.

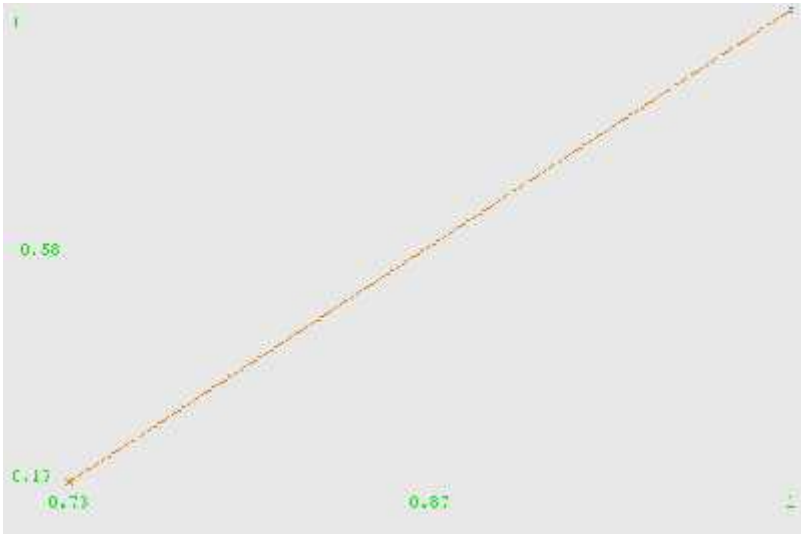

Figura 73: Representación de la curva ROC del modelo del prototipo.

De este gráfico se puede deducir que el comportamiento del modelo es relativamente significativo en cuanto a su comportamiento correcto en lo referente a la clasificación. Sin llegar a ser taxativamente lo indicado.

**4.- Análisis de las pruebas realizadas**

Los resultados obtenidos demuestran que el prototipo funciona correctamente cuando se utiliza el modelo con un valor de Gamma igual a 0,05. El cual logra clasificar el tráfico según corresponda. No sucede lo mismo para los otros valores cayéndose en fallas en la clasificación y generando falsas alarmas. Dicho valor es cercano al valor teórico calculado 0,4545, este valor teórico se logra como el inverso de la cantidad de características que se utilizarían en los vectores, que en este caso es de 22 atributos.

Una vez obtenido el valor de Gamma, se procede a realizar el análisis de data en tiempo real, generando tanto tráfico normal con algunos ataques y verificando la clasificación obtenida.

Los resultados obtenidos fueron de índole variados, por ejemplo para las pruebas a), b), c), d) y e), el prototipo logra detectar el ataque y crea la regla en el cortafuego bloqueando todo tipo de tráfico. Por otro lado, para las pruebas realizadas en f) las pruebas fallan y no se detecta el intento de intrusión con lo cual el ataque se lleva a cabo. El motivo de esta diferencia es que los primeros ataques se asocian a tráfico que corresponden a servicios no utilizados por el prototipo, por lo cual no son catalogados dentro de la gama de tráfico normal. En el caso de las pruebas f) el tráfico que se genera corresponde a un servicio prestado por el servidor, por lo cual dichas consultas se encuentran dentro de lo que se denomina un tráfico normal. Esto lleva a pensar y a comprobar que las Máquinas de Soporte Vectorial solo analizan cada vector por separado y no son capaces de asociar una conexión con la siguiente proviniendo de la misma fuente de origen. A través de las pruebas se logra demostrar que el prototipo funciona

correctamente cuando se trata de un ataque o intento de intrusión que comienza con un escaneo de puertos de servicios, en cuyo caso el prototipo bloquea automáticamente todo tráfico que provenga de dicho atacante antes de que logre recabar mayor información de las posibles vulnerabilidades que posee el sistema.

# XII. CONCLUSIONES

En el presente estudio se ha entregado una introducción al sistema de detección de intrusos (IDS), nombrando algunos modelos que se han utilizado para implementar estos sistemas. Se ha presentado un resumen de trabajos relacionados que han sido publicados y una introducción al concepto de las máquinas vectoriales. Se ha diseñado e implementado un prototipo que involucra un sistema de detección de intrusos cuya clasificación se encuentra a cargo de Máquinas de Soporte Vectorial que se implementaron con una Función Kernel de Base Radial, y cuya resolución implica una actualización automática de las reglas del Cortafuego utilizado, se realizaron las pruebas y se comprobó que el funcionamiento es correcto para cierto tipo de ataques.

La arquitectura propuesta consta de distintos módulos, cuyas funciones se han descrito anteriormente, así mismo se han detallado los distintos componentes de cada módulo y las herramientas que se utilizarán para implantarlos. Se indica a su vez las características de los distintos servidores y clientes utilizados, que por conveniencia se han virtualizado.

Como se ha explicado anteriormente, este prototipo se implementa con máquinas virtuales utilizando el programa VMWARE Workstation. En dicha plataforma se crean: un servidor WEB con sistema operativo Fedora 9, un cliente con igual sistema Operativo y un atacante con el conjunto de herramientas Backtrack 3.

En un primer paso se genera y captura una cantidad de tráfico normal y anormal que ha sido utilizado para entrenar las Máquinas de Soporte Vectorial, se utilizan distintos valores de Gamma en la función Kernel para seleccionar el que entregaba un mejor nivel de clasificación, minimizando las falsas alarmas, los resultados se han presentado anteriormente.

Una vez seleccionado el mejor valor de Gamma se procede a realizar distintas pruebas ya sean de consultas normales como de intentos de ataques o intrusión, El procedimiento para dichas pruebas es el de generar un ataque y comprobar si se había generado la regla respectiva en el Cortafuego, o por el contrario si se generaba una consulta normal se debía comprobar que no se generaran dichas reglas en el cortafuegos.

Lo anterior implica que se construye y se realizan las pruebas bajo un ambiente ideal, lo cual puede diferir en los resultados obtenidos al momento de repetir las pruebas en un servidor en producción. Para acercar un poco más la situación a un ambiente de producción se provoca una carga excesiva en el servidor y se repiten las pruebas de consultas de tráfico normal y de intentos de ataques o de intrusión.

En general y a pesar de las limitantes que se pueda tener con el prototipo propuesto, cuando se detecta un tráfico anormal se logra automatizar la creación de reglas en Cortafuego, prohibiendo el acceso a las direcciones IP que generan un posible ataque, en un mínimo tiempo, con lo cual se evita un daño mayor en la prestación de servicios.

También a través de las pruebas se logra que el administrador pueda deshacer la regla creada en el Cortafuego, a través de una consola de gestión del Cortafuego, sin exigir que el Administrador deba poseer conocimientos técnicos de la configuración del Cortafuego para realizar dicha labor.

Se puede concluir que el sistema presenta las siguientes ventajas:

- Como ventaja del sistema se desprende que entrega una solución completa para una herramienta que se podría llegar a utilizar en producción. Siendo un proyecto ambicioso al pretender ser adaptable, de fácil implementación, administración y mantención para un sistema de seguridad que incluye la actualización de las reglas del Cortafuego utilizado, contando para ello con la decisión tomada por un sistema detección de intrusos basado en las Máquinas de Soporte Vectorial, con una alta capacidad de detección.

- Una ventaja adicional y dependiendo del método utilizado para llevar a la práctica, es que se podrá realizar una fácil puesta en marcha, minimizando los tiempos de entrenamiento y conocimientos requeridos por el administrador. Las falsas alarmas son minimizadas debido a la capacidad del administrador de deshacer alguna decisión tomada por el sistema.

- Por último, se ha programado este prototipo utilizando lenguaje Shell, el cual no impone una carga excesiva al sistema al tener que traducir los comandos. Por otro lado, es un lenguaje común en el ambiente Linux, por lo que puede ser transportado a cualquier otro sistema y poder seguir operando.

- Por último, debido a las herramientas utilizadas para la implementación del prototipo, es decir, en un ambiente Shell, se logra que el proceso de análisis y clasificación de la data no provoque una carga excesiva a la máquina en que reside dicho prototipo.

Como desventaja cabe señalar los siguientes puntos:

- Corresponde a una caja negra para el administrador que solo tiene una interfaz de comunicación con el sistema pero que no sabrá realmente cuales son los motivos de una toma de decisión.

- Adicionalmente, el sistema no es capaz de detectar un intento de ataque o de intrusión cuando el tráfico generado corresponda con el servicio ofrecido por el servidor que soporta el sistema. Esto se debe a que la característica de las Máquinas de Soporte Vectorial, analiza a cada conexión en forma individual, por lo cual queda como trabajo futuro complementar este prototipo con otra herramienta que analice las conexiones como un grupo.

Por último, se ha logrado cumplir el objetivo general propuesto al inicio de esta tesis, el cual corresponde a diseñar un Prototipo de Cortafuego con actualización automática de sus reglas definidas en base a las decisiones que toma un sistema de detección de intrusos basado en anomalías, y la clasificación de los datos recibidos en un computador basada en lo resuelto por las Máquinas de Soporte Vectorial. Así también, los objetivos específicos tales como: comprender el fundamento teórico de las Máquinas de Soporte Vectorial, presentar trabajos relacionados con la implantación de distintos Sistemas de Detección de Intrusos, y utilizando Máquinas de Soporte Vectorial para la clasificación de la data, diseñar un prototipo de Cortafuego con generación de reglas en forma automática al detectar un tráfico anómalo, realizar pruebas de análisis con un prototipo de servidor y Analizar los resultados obtenidos.

## XIV. BIBLIOGRAFÍA


[1] Justo Cariacedo Gallardo, ***"Seguridad En Redes Telemáticas",*** Editorial Mcgraw Hill, 2004

[2] Latifur Khan, ***"A New Intrusion Detection System Using Support Vector Machines and hierarchical clustering",*** The VLDB Journal, Julio 2007, 16: 507 – 521

[3] William Stallings***, "Fundamentos De Seguridad En Redes: Aplicaciones Y Estandares",*** Editorial Pearson, 2° Edición, 2004.

[4] Roberto Perdisci, Giorgio Giacinto, Fabio Roli, ***"Alarm Clustering For Intrusion Detection Systems In Computer Networks",*** Department Of Electrical And Electronic Engineering, University Of Cagliari, Piazza D' Armi, 09123 Cagliari, Italy

[5] Oscar Eduardo Gualdron Guerrero, ***"Desarrollo De Diferentes Métodos De Selección De Variables Para Sistemas Multisensoriales",*** Escuela Técnica Superior De Ingeniería, Universidad Rovira I Virgili, Terragona (España), 12 de Septiembre de 2006.

[6] http://www.tcpdump.org/

[7] http://linux.die.net/man/1/inotifywait.

[8] www.gnu.org/manual/gawk/gawk.html

[9] http://www.csie.ntu.edu.tw/~cjlin/libsvm

[10] Vorgelegt Von, ***"Support Vector Learning"***, Diplom-Physiker, M.Sc. (Mathematics) Bernhard Scholkopf Aus Stuttgart, Vom Fachbereich 13, Informatik der Technischen Universitat Berlin zur Erlangung des akademischen Grades, Doktor der Naturwissenschaften, Genehmigte Dissertation, Berlin 1997

[11] Gustavo A. Betancourt, ***"Las Máquinas De Soporte Vectorial (Svms)",*** Scientia Et Technica Año Xi, No 27, Abril 2005. Utp. Issn 0122-1701

[12] Juan Manuel Górriz, ***"Nuevos Avances En Detección De Actividad De Voz Mediante Hos Y Estrategias De Optimización",*** Tesis Doctoral, Universidad de Granada, 2006

[13] Chih-Chung Chang, Chih-Jen Lin, **"LIBSVM: a Library for Support Vector Machines"**, Technical Report, Department of Computer Science, National Taiwan University, 2009.

[14] Corinna Cortes, Vladimir Vapnik, "Support-Vector Networks", AT&T Labs-Research, USA. 1995

[17] Marc Stoecklin, ***"Anomaly Detection by Finding Feature Distribution Outliers",*** IBM Zurich Research Laboratory.

[18] Shahbaz Pervez, Iftikhar Ahmad, Adeel Akram ***"A Comparative Analysis of Artificial Neural Network Technologies in Intrusion Detection Systems",*** University of Engineering and Technology, Taxila, Pakistan. Proceedings of the 6th WSEAS International Conference on Multimedia, Internet & Video Technologies, Lisbon, Portugal, September 22-24, 2006

[19] Hwanjo Yu, Jiong Yang, Jiawei Han, ***"Classifying Large Data Sets Using SVMs with Hierarchical Clusters",*** Department of Computer Science, University of Illinois, August 2003.

[20] Steve R. Gunn, ***"Support Vector Machines for Classification and Regression",*** Technical Report, Faculty of Engineering, Science and Mathematics School of Electronics and Computer Science, University of Southampton, 10 May 1998

[21] Urko Zurutuza Ortega, ***"Estado del Arte: Sistemas De Detección De Intrusos",*** Escuela Politécnica Superior, Universidad de Mondragon, Octubre de 2004

[22] José Augusto Moreno Escobar, Francisco Javier Gallegos Funes, ***"La Función De Base Radial Como Un Método Para La Detección De Microcalcificaciones En Imágenes De Mamografía",*** Sepi-Esime Zacatenco, Instituto Politécnico Nacional de México, Octubre de 2004

[23] Aleksandar Lazarevic, Levent Ertoz, ***"A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection",*** Computer Science Department, University of Minnesota, 2004

[24] Gil-Han Kim, and Hyung-woo Lee, ***"False Alarm Minimization Scheme based on Multi-Class SVM",*** IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.3B, March 2006

[25] Quang-Anh Tran, Haixin Duan, Xing Li, ***"One-class Support Vector Machine for Anomaly Network Traffic Detection",*** China Education and Research Network (CERNET), Tsinghua University, China

[26] Sandhya Peddabachigaria, Ajith Abraham, "***Modeling intrusion detection system using hybrid intelligent systems",*** Journal of Network and Computer Applications 30 (2007) 114–132

[27] Theuns Verwoerd, Ray Hunt, "***Security Architecture testing using IDS – a case study",*** Computer Communications 25 (2002) 1402-1412

**ANEXO A:**
**VALORES TASA DE VERDADEROS POSITIVOS UNITARIA (TPRu) y TASA DE FALSOS POSITIVOS UNITARIA (FPRu) PARA CADA CAMPO**

En las siguientes tablas se presentan un extracto de los valores de TPRu y FPRu, observados por cada campo en análisis.

a) Campo 1, 2, 3, 6, 7, 9:
Estos campos se han agrupado debido a que presentan el mismo comportamiento.

| TPRu | FPRu | AUC |
|---|---|---|
| 0.79384615 | 0.79692308 | 0 |

b) Campo 4:

| TPRu | FPRu | AUC |
|---|---|---|
| 0.15076923 | 0.15384615 | 0.42898935 |
| 0.15384615 | 0.15692308 | |
| 0.15692308 | 0.16 | |
| 0.16 | 0.16307692 | |
| 0.16307692 | 0.16615385 | |
| 0.16615385 | 0.16923077 | |
| 0.16923077 | 0.17230769 | |
| 0.17230769 | 0.17538462 | |
| 0.17538462 | 0.17846154 | |
| 0.17846154 | 0.18153846 | |

c) Campo 5:

| TPRu | FPRu | AUC |
|---|---|---|
| 0.00307692 | 0.00615385 | 0.47268639 |
| 0.00615385 | 0.00923077 | |
| 0.00923077 | 0.01230769 | |
| 0.01230769 | 0.01538462 | |
| 0.01538462 | 0.01846154 | |
| 0.01846154 | 0.02153846 | |
| 0.02153846 | 0.02461538 | |
| 0.02461538 | 0.02769231 | |
| 0.02769231 | 0.03076923 | |

d) Campo 8:

| TPRu | FPRu | AUC |
|---|---|---|
| 0.19076923 | 0.19384615 | 0.32434083 |
| 0.21538462 | 0.21846154 | |
| 0.75692308 | 0.76 | |
| 0.77846154 | 0.78153846 | |
| 0.78153846 | 0.78461538 | |
| 0.78461538 | 0.78769231 | |
| 0.78769231 | 0.79076923 | |
| 0.79076923 | 0.79384615 | |
| 0.80307692 | 0.80615385 | |
| 0.82153846 | 0.82461538 | |
| 0.82769231 | 0.83076923 | |

e) Campo 10:

| TPRu | FPRu | AUC |
|---|---|---|
| 0.00307692 | 0.00615385 | 0.49692308 |
| 0.00615385 | 0.00923077 | |
| 0.00923077 | 0.01230769 | |
| 0.01230769 | 0.01538462 | |
| 0.01538462 | 0.01846154 | |
| 0.01846154 | 0.02153846 | |
| 0.02153846 | 0.02461538 | |
| 0.02461538 | 0.02769231 | |
| 0.02769231 | 0.03076923 | |
| 0.03076923 | 0.03384615 | |

f) Campo 11:

| TPRu | FPRu | AUC |
|---|---|---|
| 0.03076923 | 0.03384615 | 0.49339172 |
| 0.04 | 0.04307692 | |
| 0.06153846 | 0.06461538 | |
| 0.22153846 | 0.22461538 | |
| 0.96 | 0.96307692 | |
| 0.97538462 | 0.97846154 | |
| 0.98153846 | 0.98461538 | |

0.99076923 0.99384615

0.99384615 0.99692308

g) Campo 12:

| TPRu | FPRu | AUC |
|---|---|---|
| 0 | 0.99692308 | 0.00144852 |
| 0 | 0.98461538 | |
| 0 | 0.97846154 | |
| 0 | 0.96923077 | |
| 0 | 0.96 | |
| 0.91384615 | 1 | |
| 0.94153846 | 1 | |
| 0.96 | 1 | |
| 0.96923077 | 1 | |
| 0.97846154 | 1 | |
| 0.98461538 | 1 | |
| 0.99692308 | 1 | |

h) Campo 13:

| TPRu | FPRu | AUC |
|---|---|---|
| 0 | 1 | 0.02201183 |
| 0 | 0.99692308 | |
| 0 | 0.99384615 | |
| 0 | 0.98769231 | |
| 0 | 0.98153846 | |
| 0.98153846 | 1 | |
| 0.98769231 | 1 | |
| 0.99384615 | 1 | |
| 0.99692308 | 1 | |
| 1 | 1 | |

i) Campo 14:

| TPRu | FPRu | AUC |
|---|---|---|
| 0 | 0.99384615 | 0.02201183 |
| 0 | 0.99076923 | |
| 0 | 0.98769231 | |
| 0 | 0.98153846 | |
| 0.95692308 | 1 | |
| 0.96615385 | 1 | |

| | |
|---|---|
| 0.98153846 | 1 |
| 0.98769231 | 1 |
| 0.99076923 | 1 |
| 0.99384615 | 1 |

j) Campo 15:

| TPRu | FPRu | AUC |
|---|---|---|
| 0.00615385 | 0.00923077 | 0.23750059 |
| 0.00923077 | 0.01230769 | |
| 0.01230769 | 0.01538462 | |
| 0.01846154 | 0.02153846 | |
| 0.02153846 | 0.02461538 | |
| 0.02461538 | 0.02769231 | |
| 0.03076923 | 0.03384615 | |
| 0.66461538 | 0.66769231 | |
| 0.67076923 | 0.67384615 | |
| 0.67384615 | 0.67692308 | |
| 0.68923077 | 0.69230769 | |

k) Campo 16:

| TPRu | FPRu | AUC |
|---|---|---|
| 0 | 1 | 0.00146272 |
| 0 | 0.99076923 | |
| 0 | 0.98461538 | |
| 0 | 0.97846154 | |
| 0.97846154 | 1 | |
| 0.98461538 | 1 | |
| 0.99076923 | 1 | |
| 1 | 1 | |

l) Campo 17:

| TPRu | FPRu | AUC |
|---|---|---|
| 0.00307692 | 0.00615385 | 0.46969941 |
| 0.00615385 | 0.00923077 | |
| 0.00923077 | 0.01230769 | |
| 0.01230769 | 0.01538462 | |

| | |
|---|---|
| 0.91692308 | 0.92 |
| 0.92 | 0.92307692 |
| 0.92307692 | 0.92615385 |
| 0.93230769 | 0.93538462 |
| 0.94769231 | 0.95076923 |
| 0.96923077 | 0.97230769 |

m) Campo 18:

| TPRu | FPRu | AUC |
|---|---|---|
| 0.00307692 | 0.00615385 | 0.23751479 |
| 0.00615385 | 0.00923077 | |
| 0.00923077 | 0.01230769 | |
| 0.01230769 | 0.01538462 | |
| 0.01538462 | 0.01846154 | |
| 0.64923077 | 0.65230769 | |
| 0.65230769 | 0.65538462 | |
| 0.65538462 | 0.65846154 | |
| 0.65846154 | 0.66153846 | |
| 0.68923077 | 0.69230769 | |

n) Campo 19:

| TPRu | FPRu | AUC |
|---|---|---|
| 0 | 0.99692308 | 0.02201183 |
| 0 | 0.98769231 | |
| 0 | 0.98153846 | |
| 0 | 0.97230769 | |
| 0.96615385 | 1 | |
| 0.97230769 | 1 | |
| 0.98153846 | 1 | |
| 0.98769231 | 1 | |
| 0.99692308 | 1 | |

o) Campo 20:

| TPRu | FPRu | AUC |
|---|---|---|
| 0.00307692 | 0.00615385 | 0.23751479 |
| 0.00615385 | 0.00923077 | |
| 0.00923077 | 0.01230769 | |
| 0.01230769 | 0.01538462 | |
| 0.01538462 | 0.01846154 | |
| 0.68 | 0.68307692 | |
| 0.68307692 | 0.68615385 | |
| 0.68615385 | 0.68923077 | |
| 0.68923077 | 0.69230769 | |

p) Campo 21:

| TPRu | FPRu | AUC |
|---|---|---|
| 0 | 0.99076923 | 0.00145799 |
| 0 | 0.98461538 | |
| 0 | 0.97846154 | |
| 0 | 0.96923077 | |
| 0 | 0.91076923 | |
| 0.90461538 | 1 | |
| 0.91076923 | 1 | |
| 0.96923077 | 1 | |
| 0.97846154 | 1 | |
| 0.98461538 | 1 | |
| 0.99076923 | 1 | |

q) Campo 22:

| TPRu | FPRu | AUC |
|---|---|---|
| 0 | 1 | 0.00146272 |
| 0 | 0.99076923 | |
| 0 | 0.98461538 | |
| 0 | 0.97846154 | |
| 0 | 0.97538462 | |
| 0 | 0.96923077 | |
| 0.97538462 | 1 | |
| 0.97846154 | 1 | |
| 0.98461538 | 1 | |
| 0.99076923 | 1 | |
| 1 | 1 | |

**ANEXO B**
**PROGRAMAS**

## MODULO RETROALIMENTACIÓN

### Archivo Admin

```
#!/bin/sh
iptables -L > temporal
grep "DROP" temporal| sed -e '/^$/d'|awk '{ print $4;}'>temporal2

#chmod 777 temporal
#./temporal
#service iptables save
#service iptables restart
#iptables -L
#rm -rf temporal

sed = temporal2 | sed 'N;s/\n/)\t/'
while [ true ]
do
clear
archivo=temporal2

#$(wc -l $archivo|awk '{print $1}')
echo
echo "Direcciones IP Bloqueadas "$(wc -l $archivo|awk '{print $1}')
echo "--------------------------"
cat temporal2|awk '{
print NR")           "$0;
}'
echo "s)  Salir"
echo
echo "Seleccione opcion y pulse intro: "; read lala
if [ $lala == "s" ]
then
clear
break
fi
if [ $lala  -le $(wc -l $archivo|awk '{print $1}') ]
```

```
then
echo "iptables -D INPUT "$lala > temporal
chmod 777 temporal
./temporal
service iptables save
#service iptables restart
#iptables -L
rm -rf temporal
clear
echo "********************** Borrada la regla *****************";
echo
echo
echo "********************** Presione Intro para Salir **********"; read
clear;
break;
else
echo "Opcion erronea, pulse intro para continuar..."; read
continue
fi
done
exit 0
```

## MODULO SENSOR

### Archivo Sensor

```
#!/bin/sh
#echo "********************************** Iniciando ************************************";
tcpdump -c 1 -x > logeo | while inotifywait -e modify /root/pruebas/logeo;
do
#echo "************listando*****************************";
exit;
done
# cat logeo | awk '{ split($3, a, "."); print a[1]"." a[2]"."a[3]"."a[4];exit;}'> ip_origen
grep "IP" logeo| sed -e '/^$/d' > protocolo
if [ -s protocolo ]
```

```
then
 cat logeo | awk '{ split($3, a, "."); print a[1]"." a[2]"."a[3]"."a[4];exit;}'> ip_origen
grep -v "IP" logeo | sed -e '/^$/d' > logeo2;
#else
#exit
fi
cat logeo2|awk '{
x=1
while (x <= NF) {
if (x==1 || x==10 || x== 19 || x==28 || x==37 || x==45) {
}
else {
printf $x;
printf " ";
}
x++
}
}'> logeo3
cat logeo3|awk '{
x1="0x"$1;
x2="0x"$2;
x3="0x"$3;
x4="0x"$4;
x5="0x"$5;
x6="0x"$6;
x7="0x"$7;
x8="0x"$8;
x9="0x"$9;
x10="0x"$10;
x11="0x"$11;
x12="0x"$12;
x13="0x"$13;
x14="0x"$14;
x15="0x"$15;
x16="0x"$16;
x17="0x"$17;
x18="0x"$18;
```

```
x19="0x"$19;
x20="0x"$20;
x21="0x"$21;
x22="0x"$22;
x23="0x"$23;
x24="0x"$24;

y1a= int(strtonum(x1)/4096);
y1b=int((strtonum(x1)- y1a*4096)/256);
y1c=strtonum(x1)-y1a*4096-y1b*256;
y2=strtonum(x2);
y3=strtonum(x3);
y4a= int(strtonum(x4)/8192);
y4b= int (strtonum(x4) - y4a*8192);
y5a= int(strtonum(x5)/256);
y5b= int (strtonum(x5) - y5a*256);
y6=strtonum(x6);

y7=strtonum(x7);
y8=strtonum(x8);
if (y5b=="17") {
y9=strtonum(x9)
y10=strtonum(x10)
y11=0
y12=0
y13=strtonum(x11)
y14=0
y15=0
y16=0
y17=strtonum(x12)
y18=0
};
if (y5b==6) {
y9=strtonum(x9)
y10=strtonum(x10)
y11=strtonum(x11)*65536+ strtonum(x12);
y12=strtonum(x13)*65536+ strtonum(x14);
```

```
y13=strtonum(x15)
y14=strtonum(x16)*256 + int(strtonum(x17)/256);
y15=(strtonum(x17)-strtonum(x17)*256)*65536+strtonum(x18);
y16=strtonum(x19)*65536+ strtonum(x20);
y17=strtonum(x21)*65536+ strtonum(x22);
y18=strtonum(x23)*65536+ strtonum(x24);

};

print
"1:"y1a,"2:"y1b,"3:"y1c,"4:"y2,"5:"y3,"6:"y4a,"7:"y4b,"8:"y5a,"9:"y5b,"10:"y6,"11:"y7,"12:"y8,"13:"y9,"14:"y10,"1
5:"y11,"16:"y12,"17:"y13,"18:"y14,"19:"y15,"20:"y16,"21:"y17,"22:"y18;
exit;
}' > logeo4

#echo "********************** Generacion de Vector Terminado ************************"
#rm -rf protocolo
#rm -rf logeo
#rm -rf logeo2
#rm -rf logeo3
#cat logeo4
#echo "************************ IP Origen ********************************************"
#cat ip_origen
#echo "***************************** comprobando *************************"
./clasificacion

#echo "************************ FIN ************************************************"
```

## MODULO ANÁLISIS Y ACTUALIZACIÓN DE REGLAS

### Archivo Clasificación

```
#!/bin/sh
```

```
svm-scale -r  training/entrenamiento.scale logeo4 > logeo4.scale
#svm-predict logeo4.scale  training/captura.001.model logeo4.001
#echo "**********  001 ********** ";
#cat logeo4.001
svm-predict logeo4.scale  training/captura.005.model logeo4.005
#echo "**********  005 ********** ";
#cat logeo4.005
#svm-predict logeo4.scale  training/captura.01.model logeo4.01
#echo "**********  01 ********** ";
#cat logeo4.01
#svm-predict logeo4.scale  training/captura.05.model logeo4.05
#echo "**********  05 ********** ";
#cat logeo4.05
iptables -L > temporal3
grep "DROP" temporal3| sed -e '/^$/d'|awk '{ print $4;}' > temporal2
grep "REJECT" temporal3| sed -e '/^$/d'|awk '{ print $4;}'>> temporal2
grep "ACCEPT" temporal3| sed -e '/^$/d'|awk '{ print $4;}'>> temporal2
comm -12 ip_origen temporal2 > lala
if [ -s lala ]
then
rm -rf lala
exit
fi
grep "22222" logeo4.005| sed -e '/^$/d' > trafo
if [ -s trafo ]
then

cat ip_origen | awk '{ print "iptables -I INPUT -s ",$1," -j DROP";}'>temporal
chmod 777 temporal
./temporal
#service iptables save
#service iptables restart
#iptables -L
fi
```