

**Universidad de Granada**

**ETS Ingenierías Informática y de Telecomunicación**

**Departamento de Arquitectura y Tecnología de los  
Computadores.**



**Tesis Doctoral**

**“Análisis Estadístico de Distintas Técnicas de  
Inteligencia Artificial en Detección de Intrusos”**

**Hind Tribak**

**Directores de Tesis:**

**Dr. Ignacio Rojas**

**Dr. Olga Valenzuela**

**Dr. Héctor Pomares**

**Febrero 2012.**

Editor: Editorial de la Universidad de Granada  
Autor: Hind Tribak  
D.L.: GR 1901-2012  
ISBN: 978-84-9028-099-7



## **Agradecimientos**

Me gustaría dar las gracias a todas las personas que saben el trabajo que me ha costado llevar a cabo esta tesis.

A mi director de Tesis D. Ignacio Rojas por su apoyo, ánimos y por haberme guiado por un largo camino, así como a Héctor. Por último a la Dra. Olga Cansino por su asesoramiento, trabajo, soporte y apoyo en la parte estadística, a todos ellos muchas gracias.

A Buenaventura por tu apoyo moral y ánimos en los buenos momentos y en los de bajón.

A mi familia, especialmente a mis padres y hermanos por haberme aguantado y soportado, gracias! A mi madre por estar siempre ahí a cambio de nada.

A ti papá gracias por haberme animado a continuar, por tus constantes ánimos, por ser una inspiración para mí, por haberme enseñado a luchar y a levantarme después de una batalla. Eres el espejo en el que me quiero reflejar y mi orgullo y por muchas gracias que pueda darte jamás llegaré a poder agradecerte tu gran sacrificio por el cual hoy me encuentro donde estoy.

Y por último a mi abuelo que se fue antes de poder ver este momento....

Gracias a todos.



## **Resumen**

Hoy por hoy la seguridad en redes informáticas es de vital importancia debido al gran volumen de datos que se manejan. Una de las herramientas de seguridad de las que disponen las grandes compañías son los llamados Sistemas de Detección de Intrusos o IDS.

Se han llevado a cabo varias implementaciones de IDS y las que más han triunfado hoy en día son los llamados IDS basados en uso indebido un ejemplo de ello son los antivirus. El otro tipo basado en anomalías, los cuales construyen un modelo del sistema basado en técnicas de Inteligencia Artificial para la detección de un ataque, no han corrido la misma suerte debido a que los fabricantes todavía no se fían de su comportamiento debido a la alta tasa de falsos positivos. Los sistemas basados en anomalías su estudio, se limita al ámbito académico aunque ofrecen un potencial mucho más grande y fuerte que los sistemas basados en uso indebido.

Para ello en esta tesis se pretende realizar un estudio estadístico de las diferentes técnicas de clasificación basadas en Inteligencia Artificial aplicadas a la detección de intrusos, bajo distintas perspectivas tales como son la discretización de datos y la selección de características, técnicas de reducción de datos que cada día van tomando más fuerza.



# CONTENIDO

---

---

Capítulo 1 – Introducción .....	7
1.1.    Objetivos y Organización de la Tesis .....	8
Capítulo 2 - Seguridad Informática .....	11
2.1    Definición .....	11
2.2    Elementos de la Seguridad Informática .....	11
2.3    Amenazas .....	14
2.4    Clasificación de los Ataques Informáticos .....	17
Capítulo 3 - Sistemas de detección de intrusos .....	21
3.1    Introducción .....	21
3.2    Clasificación de los IDS .....	22
3.3    Estrategia de Análisis de los IDS .....	23
3.1.1 Estrategia de Análisis: Uso Indebido.....	24
3.1.2 Anomalía .....	25
Capítulo 4 - Data Mining.....	27
4.1    Introducción .....	27
4.2    Funciones de los Modelos.....	28
4.3    Aprendizaje Automático: Aprendizaje Supervisado.....	29
4.4    Vecino más Cercano .....	31
4.4.1    Vecino más cercano e IDS.....	33
4.5    Clasificadores Bayesianos .....	33
4.5.1    Naive Bayes.....	34
4.5.2    Redes Bayesianas .....	35
4.5.3    Clasificadores Bayesianos e IDS.....	37
4.6    Árboles de Decisión.....	38
4.6.1    Algoritmo ID3 .....	39



4.6.2	Algoritmo C4.5.....	41
4.6.3	Algoritmo CART.....	43
4.6.4	Random Forest.....	44
4.6.5	Algoritmo Naive Bayes Tree.....	45
4.6.6	Árboles de Decisión e IDS .....	46
4.7	Inducción de Reglas.....	47
4.7.1	Algoritmo RIPPER “Repeated Incremental Pruning to Produce Error Reduction”.....	48
4.7.2	Algoritmo PARTIAL Decision Tree : “PART” .....	48
4.7.3	Inducción de Reglas e IDS .....	51
4.8	Lógica Difusa.....	51
4.8.1	Algoritmo Fuzzy Unordered Rule Induction Algorithm .....	52
4.8.2	Lógica Difusa e IDS .....	53
4.9	Algoritmos Genéticos .....	54
4.9.1	Algoritmos Genéticos e IDS.....	56
4.10	Sistema Inmune Artificial .....	57
4.10.1	Principio de Selección Clonal.....	57
4.10.2	Algoritmo de Selección Clonal :Clonalg.....	58
4.10.3	Sistema Inmune Artificial e IDS. ....	61
4.11	Máquinas de Soporte de Vectores “SVM”.....	62
4.11.1	Normalización. ....	66
4.11.2	SMO Mínima Secuencia de Optimización. ....	66
4.11.3	C-SVC. ....	66
4.11.4	SVM e IDS .....	67
4.12	Redes Neuronales.....	68
4.12.1	Arquitecturas de redes neuronales.....	68
4.12.2	Perceptrón Multicapa.....	69

4.12.3	Redes de función de Base Radial .....	72
4.12.4	Redes Neuronales e IDS .....	74
4.13	Modelos Ocultos de Markov .....	75
4.13.1	Arquitectura HMM. ....	76
4.13.2	Modelos Ocultos de Markov discretos .....	77
4.13.3	Modelos de Markov e IDS. ....	82
4.14	Discretización.....	82
4.15	Selección de Atributos .....	86
4.15.1	Selección por Correlation Based Filter “CFS” .....	88
4.15.2	Selección Por Consistency Based Filter “CNS” .....	89
Capítulo 5 - Estudio Experimental .....		91
5.1	Primer estudio a nivel 2 Categorías: “Normal” y “Ataque” .....	98
5.2	Primer estudio a nivel de 5 Categorías: “Dos”, “Probe”, “R2L” y “U2R. ....	100
5.3	Tercer Estudio: A Nivel de Ataque.....	103
5.4	Construcción de Modelos. ....	105
Capítulo 6 - Análisis estadístico ANOVA.....		109
6.1	Estudio estadístico de factores en detección de intrusos. ....	109
6.2	Experimentos donde sólo existen dos categorías en la variable de salida de clasificación: Ataque y Normal .....	111
6.2.1	Análisis estadístico del error global con dos categorías en la variable de salida	111
6.2.2	Análisis estadístico del tiempo de computación con dos categorías en la variable de salida .....	117
6.3	Experimentos donde existen cinco categorías en la variable de clasificación.	122
6.3.1	Análisis estadístico del error global con cinco categorías en la variable de salida	122

6.3.2	Análisis estadístico del tiempo de computación con cinco categorías en la variable de salida .....	128
6.4	Experimentos donde existen veinte categorías en la variable de salida de clasificación. Estudio a nivel de ataque.....	133
6.4.1	Análisis Estadístico del error global con 20 categorías en la variable de salida	133
6.4.2	Análisis estadístico del tiempo de computación con 20 categorías en la variable de salida. ....	137
6.5	Análisis de hipótesis .....	141
6.6	Conclusión .....	144
6.7	Trabajo Futuro .....	150
	Referencias .....	151
	Artículos .....	167
	Apéndice A	
	Apéndice B	

## Figuras presentadas en el documento

Figura 1 Gráfica que muestra n° incidentes reportados al CERT.....	16
Figura 2 Sofisticación de los ataques vs. conocimiento técnico del intruso.....	17
Figura 3 Clasificación de los IDS.....	23
Figura 4 Esquema general de un Sistema de Detección de Intrusiones. ....	23
Figura 5 IDS basado en anomalías .....	25
Figura 6 Estructura de un Clasificador Naive Bayes.....	35
Figura 7 Estructura TAN .....	37
Figura 8 NBTree con un nodo de decisión ( $X_2$ ) y 2 clasificadores NB como hojas.....	45
Figura 9 Ejemplo Algoritmo Part .....	50
Figura 10 Algoritmo Genético.....	56
Figura 11 Selección Clonal.- Algoritmo Clonalg.....	59
Figura 12 Clasificación de un conjunto de datos con SVM lineal .....	62
Figura 13 Clase linealmente Separable.	
Figura 14 Clase no linealmente Separable.....	69
Figura 15 Perceptrón con una capa oculta.....	70
Figura 16 Arquitectura típica de una RBF .....	72
Figura 17 Modelo de Urna .....	75
Figura 18 Estructura HMM Ergódico.....	77
Figura 19 Ataques/categoría/N° de registros en la Base de datos. ....	95
Figura 20 Atributos Seleccionados por los distintos métodos	
Figura 21 Selección Atributos Filtro CFS	
Figura 22 Atributos Filtro CNS	
Figura 23 Atributos Wrapper C4.5	
Figura 24 Atributos Wrapper Naive Bayes .....	98
Figura 25 Conjunto balanceado con 2 Categorías .....	99
Figura 26 Conjunto Sin Discretizar .....	99
Figura 27 Discretización Fayyad & Irani .....	100
Figura 28 Discretización Intervalo Igual de Frecuencias (n =100) .....	100
Figura 29 Presencia de ataques en el conjunto de datos.....	101
Figura 30 Presencia de ataques en el conjunto de datos por categorías .....	101
Figura 31 Conjunto de datos Sin Discretizar.....	102

Figura 32 Conjunto de datos Discretizacion Fayyad & Irani .....	102
Figura 33 Conjunto de datos Discretizacion Intervalo Igual de Frecuencias (n =100) .....	102
Figura 34 Elección del Umbral httptunnel = 133 .....	103
Figura 35 Conjunto de ataques balanceado .....	103
Figura 36 Fayyad	
Figura 37 Intervalo Igual Frec.....	104
Figura 38 Representación gráfica del resultado tras aplicar Discretización de Fayyad & Irani.....	104
Figura 39 Representación gráfica del resultado tras aplicar Discretización Igual Intervalo de Frec.....	105
Figura 40 Sin Discretizar	
Figura 41 Fayyad	
Figura 42 Intervalo Igual Frec.....	105
Figura 43 Algoritmos aplicados. ....	106

## CAPÍTULO 1 – INTRODUCCIÓN

---

Desde su invención hasta nuestros días, el número de ordenadores ha ido creciendo hasta consolidarse como un instrumento casi imprescindible en la vida cotidiana del hombre. Su versatilidad, potencia de cálculo y cada vez más fácil manejo hacen de ellos una herramienta muy importante en gran variedad de actividades, desde la científica a la lúdica. Con la posibilidad de interconectar múltiples ordenadores formando redes, surgieron nuevos retos y aplicaciones.

La red ARPANET, creada por el gobierno estadounidense en 1969 para actividades de desarrollo y defensa, sería la precursora de la que hoy conocemos como Internet. En aquel entorno, la seguridad era mínima. Se trataba de una red compuesta por una pequeña comunidad cuyos miembros eran de confianza. La mayoría de los datos que se intercambiaban no eran confidenciales, y muchos usuarios se conocían. La ARPANET original evolucionó hacia Internet. Internet se basó en la idea de que habría múltiples redes independientes, de diseño casi arbitrario, empezando por ARPANET como la red pionera de conmutación de paquetes, pero que pronto incluiría redes de paquetes por satélite, redes de paquetes por radio y otros tipos de red. Internet como ahora la conocemos encierra una idea técnica clave, la de arquitectura abierta de trabajo en red.

Internet ha supuesto una revolución sin precedentes en el mundo de la informática y de las comunicaciones. Es difícil imaginarse hoy algún banco, hospital, o gran superficie comercial en un país desarrollado, que no mantenga los datos de sus clientes o hagan sus transacciones de forma electrónica. Hoy en día los bancos hacen uso de redes para efectuar sus operaciones financieras, los hospitales tienen los historiales de sus pacientes en bases de datos, y muchos comercios están presentes en Internet, de forma que cualquier usuario del planeta puede tanto escoger el producto que desea como pagarlo a través de la red. Los datos que manejan este tipo de empresas deben mantenerse a salvo de cualquier intruso a toda costa. La seguridad en este tipo de empresas tiene una importancia crítica. La información es el activo más importante en los negocios actuales, de hecho los ciber-ataques a las grandes compañías siguen siendo un gran problema en el mundo empresarial, ya que llegan a causarles gastos de millones

de dólares al año para luchar contra las amenazas de la red. Así pues hoy en día salvaguardar la información en la red y tener una buena política de seguridad se vuelve primordial y de suma importancia teniendo en cuenta que los ataques son difíciles de prevenir con los cortafuegos, las políticas de seguridad, u otros mecanismos, porque el software de aplicación está cambiando a un ritmo rápido, y este rápido ritmo a menudo conduce a un software que contiene fallos desconocidos o errores.

### 1.1. Objetivos y Organización de la Tesis

La seguridad informática consiste en asegurar que los recursos tales como las metodologías, planes, políticas, documentos, programas o dispositivos físicos, encaminados a lograr que los recursos de cómputo disponibles en una organización o ambiente dado, sean accedidos única y exclusivamente por quienes tienen la autorización para hacerlo y dentro de los límites de su autorización. Para ello disponemos de varias herramientas que nos ayudan a mantener la seguridad de una organización red o recurso cualquiera, entre estas herramientas están los llamados sistemas de detección de intrusos. Los sistemas de intrusos basados en red, los cuales monitorizan el tráfico de una red, se clasifican dependiendo del tipo de análisis que lleven a cabo en dos tipos. Por un lado están los de uso indebido que requieren de un sistema de apoyo, como es una base de datos, éstos necesitan de un mantenimiento regular, como son las actualizaciones periódicas, ejemplo de ello son los antivirus. Por otra parte nos encontramos con los llamados sistemas basados en anomalías los cuales no cuentan con ningún soporte, puesto que aprenden a modelar, gracias a un algoritmo de inteligencia artificial, el comportamiento normal del sistema. Éstos algoritmos les permiten aprender y por ello perfilar una conducta de actividad normal y todo lo que se desvíe de esa conducta es reportado como una anomalía o intrusión.

Los sistemas de detección de intrusos basados en anomalías no están tan desarrollados por los fabricantes debido a su baja fiabilidad frente a los sistemas de detección basados en el uso indebido.

En esta tesis se pretende realizar una comparativa estadística global de distintos algoritmos de Inteligencia Artificial que se aplican en detección de intrusos y establecer qué algoritmo es más ventajoso en determinadas condiciones frente a otros, siendo estas condiciones el tipo de discretización, y la selección de atributos. Básicamente

analizaremos estas condiciones, las cuales serán factores que influirán en el acierto o error de clasificación del modelo y en su tiempo de construcción.

El criterio que se ha seguido para seleccionar los distintos algoritmos de aprendizaje es el buen resultado que han dado en otros ámbitos de estudio.

Estos algoritmos se entrenarán y se evaluarán utilizando el conjunto de datos NSL [NSL09], el cual es una base de datos con miles de patrones de firmas de ataques así como de conexiones normales, y a la vez es una mejora de los datos del concurso KDD cup'99[Kdd99]. En KDD-99 se utilizó una versión reducida de la amplia variedad de intrusiones militares simuladas en un entorno de red, proporcionadas por DARPA Intrusion Detection Program Evaluation en 1998. En este conjunto de datos cada registro de conexión está compuesto de 42 atributos, lo que supone unos 100 bytes por registro.

Para llevar a cabo este estudio se ha utilizado la selección de atributos, proceso que consiste en seleccionar a partir de los datos de entrada un subconjunto óptimo de características de una base de datos para reducir su dimensionalidad, eliminar ruido y mejorar el desempeño de un algoritmo de aprendizaje. Sería muy interesante el estudiar si un modelo sigue siendo bueno o desmejora si se entrena y evalúa con pocos atributos reduciendo así la dimensionalidad del problema y manteniendo las prestaciones del algoritmo. Para ello someteremos a los datos a diferentes técnicas de selección de atributos que se tratarán más adelante.

Además de la selección de atributos se ha utilizado la discretización que es de especial importancia en Inteligencia Artificial, pues permite que muchos algoritmos de aprendizaje ideados para funcionar con atributos nominales o categóricos puedan también utilizarse con conjuntos de datos que incluyen valores numéricos, algo esencial en la resolución de problemas reales. Nuestro conjunto de datos como se verá más adelante cuenta con que la mayoría de los atributos es de tipo continuo, y algunos algoritmos de aprendizaje operan exclusivamente con espacios discretos. Se llevará a cabo dos tipos de discretizaciones distintas, una supervisada y otra no supervisada que serán explicadas más adelante. Como pasaba con la selección de atributos gracias a la discretización estudiaremos si las prestaciones de un clasificador mejoran o desmejoran así como cual de ambos métodos de discretización resulta ser mejor que otro.



Tomando la discretización, el filtro y el tipo de algoritmo como factores que influyen en el desempeño de un modelo basado en anomalías, analizaremos el comportamiento de este modelo, acierto en clasificación y tiempo de construcción, desde tres puntos de vista distintos: sistema de clasificación binaria (ataque/no-ataque), sistema de clasificación en 5 categorías (DoS, Probe, R2L, U2R), sistema de clasificación a nivel de ataque (veinte ataques diferentes).

Esta tesis se va a organizar de la siguiente manera, en el capítulo 2 se estudiará la seguridad informática, los elementos que la componen así como las amenazas y clasificación de los ataques informáticos. En el capítulo 3 se tratarán los sistemas de detección de intrusos y se expondrá los tipos que existen y nos centraremos en la estrategia de análisis en las que se basan. En el capítulo 4 se expondrán los algoritmos de aprendizaje inteligente más usuales y frecuentes, utilizados y referenciados en la bibliografía. Se explicarán las diferentes técnicas o algoritmos que se aplicarán en nuestro estudio experimental como son los clasificadores bayesianos, modelos de Markov, máquinas de soporte vectorial, lógica difusa, árboles de decisión etc..y al final de cada apartado se comentará un breve estado del arte de la técnica en cuestión y su aplicación a los sistemas de detección de intrusos. Los dos últimos apartados de este capítulo tratarán sobre las diferentes técnicas de discretización y de selección de atributos escogidas para el procedimiento experimental. En el capítulo 5 se explicará el procedimiento experimental que se ha llevado a cabo, así como el conjunto de datos utilizado y los 3 casos o perspectivas de estudio a las que se ha sido sometido este conjunto de datos así como los algoritmos y sus variantes que se utilizarán para su evaluación.

Por último en el capítulo 6 partiendo de las tablas que se construirán en la fase experimental se realizará el estudio estadístico ANOVA y se ofrecerán las conclusiones obtenidas.

En los apéndices de esta tesis se podrá encontrar las matrices de confusión de cada clasificador así como las tablas obtenidas con los resultados como son, el tiempo de construcción del modelo y el acierto global del clasificador así como otros aciertos dependiendo del caso de estudio.

## CAPÍTULO 2 - SEGURIDAD INFORMÁTICA

---

### 2.1 Definición

La *seguridad informática* consiste en asegurar que los recursos del sistema de información (material informático o programas) de una organización, sean utilizados de la manera que se decidió y que el acceso a la información allí contenida, así como su modificación, sólo sea posible a las personas que se encuentren acreditadas y dentro de los límites de su autorización.

Podemos definir la Seguridad Informática como el cumplimiento de *confidencialidad, integridad y disponibilidad* en un sistema informático [Rus91]. La confidencialidad requiere que la información sea accesible únicamente por aquellos que estén autorizados, la integridad que la información se mantenga inalterada ante accidentes o intentos maliciosos, es decir la información sólo puede ser modificada por quien está autorizado y de manera controlada y la disponibilidad significa que el sistema informático se mantenga trabajando sin sufrir ninguna degradación en cuanto a accesos y provea los recursos que requieran los usuarios autorizados cuando éstos los necesiten.

### 2.2 Elementos de la Seguridad Informática

*Políticas de Seguridad:-* En cualquier entorno en el que tengamos datos o información que hay que proteger no sólo son importantes las herramientas de las que disponemos para proteger dicha información sino también el disponer de una buena política de Seguridad.

Las políticas de seguridad informática surgen como una herramienta organizacional para concienciar a los colaboradores de la organización sobre la importancia y sensibilidad de la información y servicios críticos que permiten a la empresa crecer y mantenerse competitiva. No se puede considerar que una política de seguridad informática es una descripción técnica de mecanismos, ni una expresión legal que involucre sanciones a conductas de los empleados, es más bien una descripción de lo que deseamos proteger y él por qué de ello y cómo lograrlo, pues cada política de seguridad es una invitación a cada uno de sus miembros a reconocer la información

como uno de sus principales activos así como, un motor de intercambio y desarrollo en el ámbito de sus negocios.

Las Políticas de Seguridad Informática deben considerar principalmente los siguientes elementos:

- Alcance de las políticas, incluyendo facilidades, sistemas y personal sobre la cual aplica.
- Objetivos de la política y descripción clara de los elementos involucrados en su definición.
- Responsabilidades por cada uno de los servicios y recursos informáticos aplicado a todos los niveles de la organización.
- Requerimientos mínimos para configuración de la seguridad de los sistemas que abarca el alcance de la política.
- Definición de violaciones y sanciones por no cumplir con las políticas.
- Responsabilidades de los usuarios con respecto a la información a la que tiene acceso.
- Las políticas de seguridad informática, también deben ofrecer explicaciones comprensibles sobre por qué deben tomarse ciertas decisiones y explicar la importancia de los recursos. Igualmente, deberán establecer las expectativas de la organización en relación con la seguridad y especificar la autoridad responsable de aplicar los correctivos o sanciones.
- Otro punto importante, es que las políticas de seguridad deben redactarse en un lenguaje sencillo y entendible, libre de tecnicismos y términos ambiguos que impidan una comprensión clara de las mismas, claro está sin sacrificar su precisión.
- Por último, y no menos importante, el que las políticas de seguridad, deben seguir un proceso de actualización periódica.

El *cifrado* es el más antiguo de entre los mecanismos de protección. Un sistema secreto se define como un conjunto de transformaciones de un espacio a otro espacio, donde cada transformación en particular corresponde a un cifrado con una llave particular [Sha49]. La criptografía es el estudio de sistemas matemáticos que involucra a dos problemas de seguridad: privacidad y autenticación [Dif76]. La Criptografía es

una rama de las matemáticas que, al orientarse al mundo de los mensajes digitales, proporciona las herramientas idóneas para solucionar los problemas relacionados con la autenticidad y la confiabilidad. El problema de la confidencialidad se vincula comúnmente con técnicas denominadas de "encriptación" y la autenticidad con técnicas denominadas "firma digital", aunque la solución de ambos, en realidad, se reduce a la aplicación de procedimientos criptográficos de encriptación y desencriptación.

Por otro lado, la mayoría de los sistemas de ordenadores proveen mecanismos de *control de accesos* en su primera línea de defensa [Lam71]. Este mecanismo únicamente limita el acceso a un objeto en el sistema, pero no modela ni restringe qué es lo que un sujeto puede hacer con el objeto en el caso de que tenga acceso a su manipulación [Den82].

El flujo de información se puede controlar para incrementar la seguridad mediante la aplicación de modelos como el de Bell y LaPadula [Bel73] para proveer *confidencialidad*, o el modelo Biba [Bib77] para proveer *integridad*. Ambos modelos son conservadores y restringen operaciones de lectura y escritura para asegurar que no se pueda comprometer la integridad y la confidencialidad de los datos de un sistema. Por ello, un sistema completamente seguro no sería de gran utilidad ya que sería demasiado restrictivo [Kum95].

Los controles de acceso y modelos de protección no son útiles ante amenazas internas. Si una contraseña es débil y se compromete, las medidas de control de acceso no pueden prevenir la pérdida o corrupción de la información a la que el usuario estaba autorizado a acceder. En general, los métodos estáticos de aseguramiento de propiedades de seguridad en un sistema o son insuficientes, o pueden resultar demasiado restrictivos para los propios usuarios.

Por otro lado también podemos encontrar mecanismos de *identificación* y *autenticación* (I&A). Estos mecanismos posibilitan la identificación adecuada de los sujetos y objetos del sistema. La identificación es la *declaración de quién es el usuario* (conocido a nivel global), mientras que autenticación es la *prueba o confirmación de esa identificación* [NSA89]. Existen tres tipos de identificación: la declaración de identidad, identidad colectiva y habilidad. Asimismo, esas identidades de los usuarios se verifican mediante tres métodos genéricos: *Lo que saben* (contraseñas, PIN,...), *lo que*

*tienen* (tarjetas magnéticas, claves electrónicas,...) y finalmente *lo que son* (autenticación biométrica como iris, huellas dactilares,...).

Por último, hay otra serie de mecanismos con el objetivo de velar por la *disponibilidad* de un sistema. Algunos de ellos actúan a modo de filtros, dejando pasar aquella información que esté autorizada, en el caso de routers (con listas de acceso o ACL) y cortafuegos. Y por último los que de alguna manera detectan amenazas, como antivirus y sistemas de detección de intrusos. Éstos últimos forman la última línea de defensa en el esquema general de protección de un sistema informático, y no sólo son útiles para detectar incidentes de seguridad, sino también intentos de romper la seguridad.

### 2.3 Amenazas

El uso creciente de los sistemas de ordenadores ha exacerbado el problema de accesos no autorizados y la manipulación de datos. El alto nivel de conectividad en el que nos encontramos no sólo proporciona acceso a gran cantidad y variedad de fuentes de datos más rápido que nunca antes, sino que lo provee desde cualquier lugar en la red [Pow95]. Desde el ataque del gusano Internet de 1988 [Spa89], ha habido una innumerable cantidad de intrusiones de red que se han saltado los mecanismos establecidos para la protección de los sistemas. Hay muchas amenazas que ya existían y las nuevas tecnologías han provocado que aparezcan nuevas, por ejemplo los móviles o los ataques a la web. Las nuevas tecnologías y plataformas ayudan a que salgan nuevos ataques. Las amenazas han existido siempre pero tal vez ahora la gente las comunica más. Los delitos en la red en estos momentos están ingresando más dinero que delitos de tráfico de drogas o de armas.

Los ataques informáticos online se han convertido en una constante. La evolución de la tecnología y la organización de grupos de 'hackers' han hecho que las amenazas de seguridad aumenten y que las autoridades tengan que actualizarse constantemente para poder hacer frente a la múltiple variedad de ataques, por ejemplo para intentar facilitar herramientas de colaboración entre cuerpos de policía de todo el mundo, Symantec ha fundado el Instituto de seguridad Norton.

Páginas web gubernamentales, financieras y empresariales de todo el mundo conocen al movimiento 'Anonymous' ya que son el blanco de sus objetivos, este grupo de hackers llevan a cabo un ataque de denegación de servicio colapsando las webs.

Grandes empresas son el objetivo de los delincuentes, ejemplo de ello en 2011 Google Inc. sufrió un importante ataque a Gmail, los atacantes intentaban hacerse con las cuentas de correos de usuarios de entre ellos estarías gente del gobierno, periodistas etc. Facebook es una de las mayores redes sociales que existen en la actualidad superando los 100 millones de usuarios registrados en 2008 y posteriormente en 2010 fue víctima de un ataque informático por el método conocido como phishing, en el que los delincuentes imitan la apariencia de esta página web para robar los datos de registro de los usuarios. En 2011 grandes compañías como son Sony reveló el coste asociado con la reparación de la brecha masiva de seguridad que expuso la información personal de más de 100 millones de usuarios de los servicios PlayStation Network y Qriocity: que le costó mínimo 171 millones de dólares. La compañía japonesa SEGA, ha reconocido haber sufrido un ataque informático a su base de datos. Se produjo el acceso no autorizado a los datos de 1,3 millones de clientes de su canal de juegos Sega Play. El episodio fue un recordatorio de los retos de cuán importante es la seguridad de los datos y un indicador de que muchas organizaciones no se están protegiendo lo suficientemente bien. Cuando se trata de todos estos problemas de seguridad, las empresas no están invirtiendo por avanzado, pero después tienen que gastar mucho dinero para arreglar las cosas.

El Instituto Nacional de Tecnologías de la Comunicación (Inteco)[Int], centro estatal especializado en seguridad en la Red y ubicado en León, a través de su Centro de Respuesta a Incidentes de Seguridad de la Información, Inteco-CERT, ha superado la cifra de 10.000 virus analizados y catalogados en su base de datos. Esta información es importante para que usuarios particulares y empresas conozcan a qué amenazas se enfrentan en internet. Cada semana, el Inteco-CERT localiza y clasifica una media de 12 códigos maliciosos, con la finalidad de conseguir una mayor seguridad en internet y de asesorar a los usuarios sobre el modo de protegerse de las diferentes amenazas que van surgiendo en la Red. El estudio de los virus identificados por el instituto refleja la evolución de la tecnología y el hecho de que los virus permanecen en el tiempo y evolucionan, adaptándose a los nuevos sistemas operativos y ampliando su campo de acción a nuevas plataformas. Así, los técnicos del Inteco han observado que cada vez

hay más virus documentados que afectan a dispositivos móviles como iPad, iPhone, Android o Symbian, entre otros, que se unen a los que atacan a sistemas operativos distintos de Windows, como Mac OS o Linux.

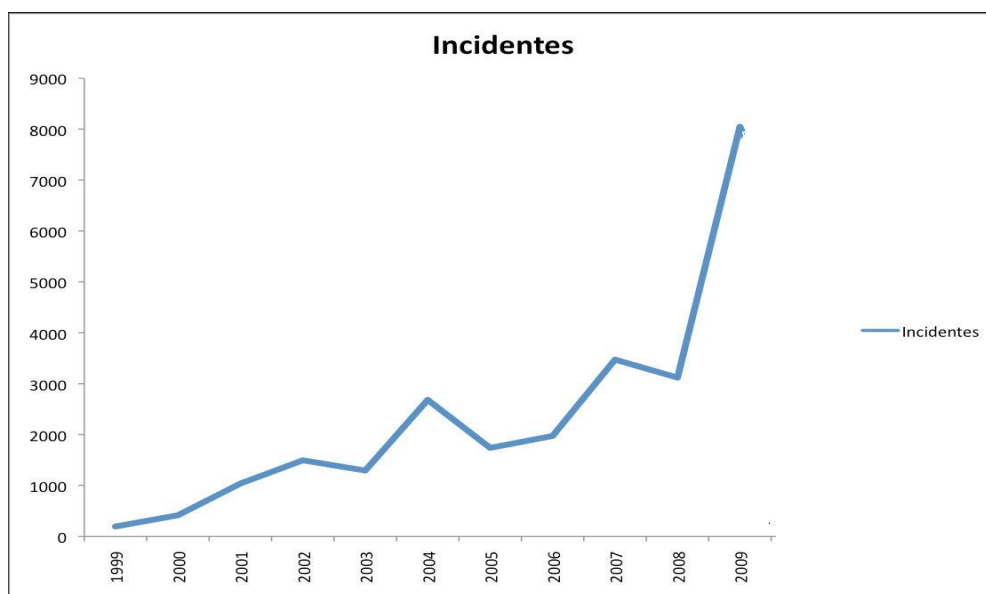


Figura 1 Gráfica que muestra nº incidentes reportados al CERT.

Otro aspecto a tener en cuenta es la dificultad que conlleva la realización de software, ya que éste es cada vez más complejo y el ciclo de vida del software se está reduciendo significativamente debido al aumento de la competitividad del mercado. Este hecho acarrea la consecuencia de realizar diseños pobres, testeado inadecuado, y por lo tanto errores en el software que se manifiestan como vulnerabilidades de seguridad.

Antes, los intrusos necesitaban de un conocimiento más profundo de las redes y las computadoras para poder lanzar sus ataques. Desgraciadamente, gracias al incremento del conocimiento sobre el funcionamiento de los sistemas, los intrusos están cada vez más preparados y lo que antes estaba accesible para sólo unos pocos (expertos), hoy en día cualquiera tiene herramientas accesibles con las que poder determinar las debilidades de los sistemas y explotarlas con el fin de obtener los privilegios necesarios para realizar cualquier acción dañina. Esto puede observarse en la siguiente gráfica:

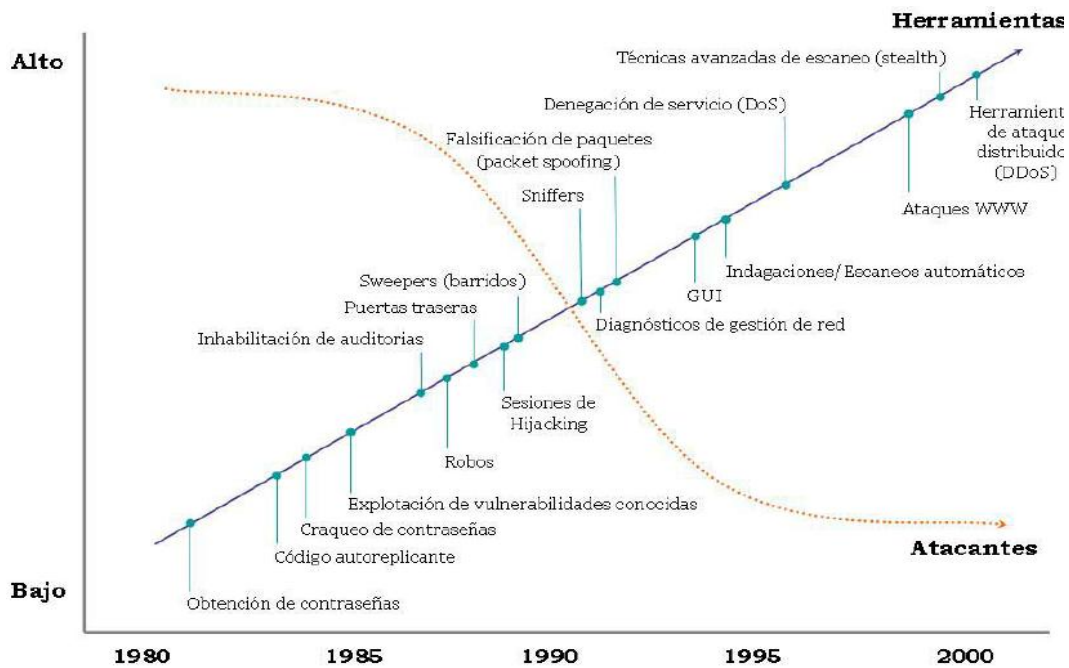


Figura 2 Sofisticación de los ataques vs. conocimiento técnico del intruso

## 2.4 Clasificación de los Ataques Informáticos

El propósito de una clasificación o taxonomía es proporcionar un medio útil y coherente de clasificar los ataques. Actualmente, los ataques a menudo se describen de forma diferente por diferentes organizaciones, dando lugar a confusión a los que en realidad es un ataque en particular. Por ejemplo, una organización puede clasificar como un ataque a un virus mientras que otra lo clasifica como un gusano. Una taxonomía permite tener un conocimiento previo que se aplicará a nuevos ataques así como proporciona una forma estructurada para el estudio de estos ataques. Se pueden encontrar multitud de trabajos referentes a la categorización y clasificación de ataques informáticos e intrusiones [Asl95][Kum95b][Lan94][Lou01][Shan04].

Uno de los primeros trabajos dedicados a categorizar diferentes aspectos de la seguridad informática, se centraba en la debilidad de los sistemas informáticos y defectos de diseño en sistemas operativos [Att76], así como en vulnerabilidades funcionales y métodos de abusos informáticos [Par75]. Varias de las taxonomías desarrolladas más tarde se enfocaban principalmente en dos aspectos: categorización de uso indebido de computadoras, y categorización de la gente que intentaba obtener acceso no autorizado a ordenadores.



En un intento anterior de describir tipos de ataques informáticos, Neumann y Parker desarrollaron el SRI Computer Abuse Methods Model [Neu89a][Neu89b][Neu95], el cual describe aproximadamente 3000 ataques y usos indebidos recogidos durante unos veinte años, y los clasifica en un árbol de nueve categorías de ataques. Lindqvist y Jonson extendieron este modelo expandiendo las categorías 5, 6 y 7 del árbol original [Lind97]. Jayaram y Morse [Jay97] también desarrollaron una clasificación de amenazas de seguridad en redes, en la que proveen cinco clases de amenazas de seguridad y dos clases de mecanismos de seguridad. Otro trabajo significativo en taxonomías de ataques fue realizado por el grupo CERIAS de Purdue University [Asl95][Kum95b][Krs98]. Inicialmente, Sandeep Kumar realizó una clasificación de intrusiones en sistemas de ordenadores UNIX basado en logs del sistema y redes de Petri coloreadas. Aslam extendió dicho estudio añadiendo una taxonomía de fallos de seguridad en sistemas UNIX. Finalmente, Iván Krsul reorganizó las dos taxonomías anteriores y proporcionó una taxonomía más compleja de ataques informáticos organizados en cuatro grupos (diseño, supuestos ambientales, fallos de codificación y errores de configuración). Richardson [Ric99][Ric01] extendió estas clasificaciones desarrollando una base de datos de vulnerabilidades para ayudar en el estudio del problema de ataques de denegación de servicio (DoS). La base de datos se pobló con 630 ataques de sitios populares donde se reportaban incidentes informáticos. Estos ataques se clasificaron dentro de las categorías correspondientes a las extensiones de la taxonomía de fallas de seguridad de Aslam y de la taxonomía de Krsul. Dentro del proyecto de evaluación de detección de intrusos DARPA (DARPA intrusion detection evaluation data sets) [DAR04], Kendall [Ken98] desarrolló una base de datos de ataques similar, que se puede encontrar en los conjuntos de datos de evaluación de detección de intrusos DARPA. En esta base de datos, utilizada actualmente como elemento evaluador y comparativo de los sistemas de detección desarrollados por los investigadores, los ataques se clasifican en cuatro grupos principales, utilizando como criterio el *tipo de ataque*:

**Denegación de Servicio (DoS):** Estos ataques tratan de detener el funcionamiento de una red, máquina o proceso; o si no denegar el uso de los recursos o servicios a usuarios autorizados [Mar01]. Hay dos tipos de ataques DoS; por un lado ataques de sistema operativo, los cuales tratan de explotar los fallos en determinados sistemas operativos y pueden evitarse aplicando los respectivos parches; y ataques de red, que

explotan limitaciones inherentes de los protocolos e infraestructuras de red. Hay varios tipos de denegación de servicio (DOS) [Ken98], algunos ataques como "mailbomb", "Neptune" o "smurf" abusan de los dispositivos legítimos. Otros como "teardrop", crean paquetes malformados que confunden a los protocolos TCP / IP en la máquina objetivo, y ésta última intentará la reconstrucción de los paquetes. Y otros "apache2" ó back" sacan provecho de los errores en la red.

**Indagación o exploración (probing):** Este tipo de ataques escanean las redes tratando de identificar direcciones IP válidas y recoger información acerca de ellas (servicios que ofrecen, sistemas operativos que usan). A menudo, esta información provee al atacante una lista de vulnerabilidades potenciales que podrían ser utilizadas para llevar a cabo ataques a los servicios y a las máquinas escogidas. Estos ataques son los más frecuentes, y a menudo son precursores de otros ataques. Un atacante con un mapa de las máquinas y servicios disponibles en una red puede utilizar esta información para encontrar todos los puntos débiles de esta última. Algunas de estas herramientas de análisis "satan, saint, mscan" permiten que incluso un hacker principiante, pueda revisar rápidamente cientos o miles de máquinas en una red.

**R2L (Remote to Local):** cuando un atacante que no dispone de cuenta alguna en una máquina, logra acceder (tanto como usuario o como root) a dicha máquina. En la mayoría de los ataques R2L, el atacante entra en el sistema informático a través de Internet. Hay varias maneras en que un atacante puede lograr su objetivo [Kendall, 99]. Algunos ataques explotan el desbordamiento de búfer causado por el software de servidor de red "imap, named, sendmail". Los ataques de " ftp\_write, xsnoop y guest" tratan de explotar la debilidad o la mala configuración de las políticas de seguridad del sistema. El ataque "xlock" utiliza ingeniería social para tener éxito, el atacante debe suplantar a los operadores humanos que proporcionan sus contraseñas de los protectores de pantalla que en realidad son caballos de Troya.

**U2R (User to Root):** Este tipo de ataque se da cuando un atacante que dispone de una cuenta en un sistema informático es capaz de elevar sus privilegios explotando vulnerabilidades en los mismos, un agujero en el sistema operativo o en un programa instalado en el sistema. Hay varios tipos de ataques U2R [Ken98] La más común es el ataque "buffer\_overflow" que se produce cuando un programa copia una gran cantidad de datos en un búfer de memoria estática sin comprobar si el tamaño de esta última es

suficiente, lo que provocará un desbordamiento. Los datos desbordados se almacenan en la pila de sobrecarga del sistema, cubriendo así las siguientes instrucciones para ser ejecutadas. Mediante la manipulación cuidadosa de los datos almacenados en la pila, un atacante puede provocar la ejecución de código en el sistema operativo que le ayudará a conseguir lo que quiere. Otra clase de ataques U2R explotan los programas que proporcionan información sobre el medio en el que se ejecutan, un buen ejemplo de este tipo de ataque es el ataque "iodamodule". Otra clase de ataques U2R explotan los programas que tienen una mala gestión de los archivos temporales. Algunos ataques U2R explotan la vulnerabilidad debido a las condiciones competitivas explotables durante la ejecución de un solo programa, dos o más programas se ejecutan simultáneamente [Gar96]. A pesar de que una programación controlada podría eliminar todas estas vulnerabilidades, tales errores están presentes en todas las versiones de UNIX y Windows de Microsoft disponibles hoy en día.

# CAPÍTULO 3 - SISTEMAS DE DETECCIÓN DE INTRUSOS

---

## 3.1 Introducción

Las primeras investigaciones sobre detección de intrusos comienzan en 1980 en un trabajo de consultoría realizado para el gobierno norteamericano por James P. Anderson [And80], quien trató de mejorar la complejidad de la auditoría y la habilidad para la vigilancia de sistemas informáticos. Anderson presentó la idea de que el comportamiento normal de un usuario podría caracterizarse mediante el análisis de su actividad en los registros de auditoría. De ese modo, los intentos de abusos podrían descubrirse detectando actividades anómalas que se desviarán significativamente de ese comportamiento normal.

Se puede definir **intrusión** como la *violación de la política de seguridad de un sistema*, o como la *materialización de una amenaza*. Heady et al. [Hea90] definen intrusión como *cualquier 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* [Powe01], aunque es bien sabido que muchas de las intrusiones proceden del interior del sistema de información. Finalmente, el NIST (National Institute of Standards and Technology) define detección de intrusos como *el proceso de monitorización de eventos que suceden en un sistema informático o red y análisis de dichos eventos en busca de signos de intrusiones*.

El primer modelo de detección de anomalías fue el propuesto por Dorothy Denning, con la idea básica de monitorizar las operaciones estándares de un sistema objetivo, observando desviaciones en su uso [Den87]. Su artículo provee un marco metodológico que más tarde inspiraría a muchos investigadores.

Entre 1988 y 1990 el Instituto de Investigación SRI International desarrolla la propuesta de Denning. De ese modo surge IDES (Intrusion Detection Expert System), un sistema experto que detecta las desviaciones a partir del comportamiento de diferentes sujetos [Lun88][Lun90]. IDES fue el primer sistema de detección de anomalías en host.

En 1988, simultáneamente, en los laboratorios Lawrence Livermore de University of California en Davis, se realiza el proyecto Haystack para las fuerzas aéreas de EE.UU. Haystack era el primer IDS que analizaba los datos de auditoría y los comparaba con patrones de ataque predefinidos [Sma88]. De este modo nacía el primer sistema de detección de usos indebidos basado en firmas, el tipo de IDS más extendido en el mercado actual.

En 1990, surgen los primeros proyectos de IDS basados en red. Todd Heberlein introduce tal idea y desarrolla NSM (Network Security Monitor) en University of California at Davis [Heb90]. En esa misma fecha, en Los Alamos National Laboratory de EEUU realizan un prototipo de un sistema experto que monitoriza la actividad de red. Su nombre es NADIR (Network Anomaly Detector and Intrusion Reporter) [Jac90].

A partir de este momento, comienzan una gran variedad de proyectos de investigación que hacen uso de diferentes técnicas y algoritmos para el análisis del comportamiento de un sistema informático.

### 3.2 Clasificación de los IDS

La clasificación o taxonomía de los sistemas de detección de intrusos ha sido tratada en numerosos trabajos, de los que destacan los de Hervé Debar [Deb99] y Stefan Axelsson [Axe00] de Chalmers University of Technology en Suecia. La clasificación más común se realiza en base a tres características funcionales de los IDS:

*Fuentes de información.* Se refiere al origen de los datos que se usan para determinar si una intrusión se ha llevado a cabo. Básicamente existen 2 tipos aquellos que obtienen sus datos de una máquina o host, y aquellos que los obtienen a partir de la monitorización de una red.

*Análisis.* Se trata del método de detección utilizado. La información recogida en el paso anterior puede ser analizada mediante dos estrategias diferentes, una basada en uso indebido y la otra basada en anomalías.

*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.

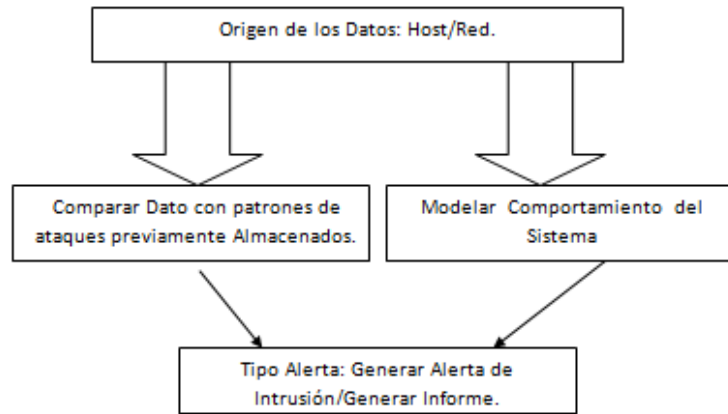


Figura 3 Clasificación de los IDS

### 3.3 Estrategia de Análisis de los IDS

Después del proceso de recopilación de información, se lleva a cabo el proceso de análisis. Los dos tipos principales de análisis son:

- **Detección de usos indebidos ("misuse detection"):** Para encontrar usos indebidos se comparan *firmas* (patrones de ataques conocidos) con la información recogida en busca de coincidencias.
- **Detección de anomalías:** Para la detección de anomalías se manejan técnicas estadísticas que definen de forma aproximada lo que es el comportamiento usual o normal.

La siguiente figura muestra un esquema general de detector de intrusiones de usos indebidos (mediante comparación de patrones) y de anomalías.

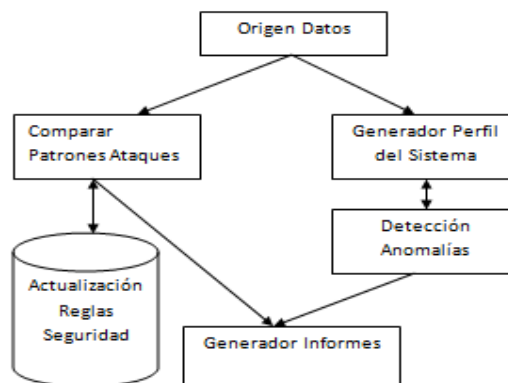


Figura 4 Esquema general de un Sistema de Detección de Intrusiones.

Nos vamos a centrar en la estrategia de análisis de los IDS, pasaremos a describir las dos estrategias básicas en las que un IDS se basa para detectar si se está llevando a cabo o no un ataque.

### 3.1.1 Estrategia de Análisis: Uso Indebido

Un IDS basado en detección de uso indebido monitoriza 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 monitorizadas 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 IDS comerciales y por la que apuestan los fabricantes.

Típicamente, un sistema de detección de uso indebido contiene dos componentes principales [Kum94]:

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

La *ventaja* de los IDS 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 IDS al sistema en el que se implanta si no queremos que se dispare el número de falsos positivos -una intrusión anómala, la actividad es no intrusiva, pero como es anómala el sistema decide que es intrusiva. Se denominan falsos positivos, porque el sistema erróneamente indica la existencia de intrusión-.

### 3.1.2 Anomalía

Consiste en la elaboración de perfiles estadísticos de comportamiento a lo largo del tiempo. Estos perfiles se construyen mediante determinados algoritmos, capaces de detectar cambios graduales en los patrones de conducta de los usuarios o anomalías.

Una anomalía se puede definir como la *discrepancia de una regla o de un uso* [Rae04]. 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 auditoría, llamadas del sistema de los procesos, etc.). Una vez definido esto, clasificará como sospechosas o intrusivas aquellas desviaciones que pueda detectar sobre el comportamiento normal. 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 significativa pueda ser considerada como evidencia de una intrusión.

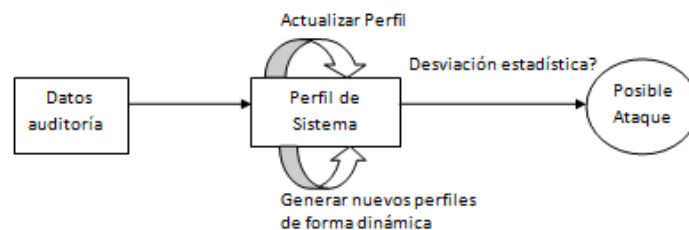


Figura 5 IDS basado en anomalías

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.

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. 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. 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. Básicamente los sistemas basados en detección de anomalías se clasifican en, sistemas basados en conocimiento por ejemplo sistemas expertos, sistemas basados en métodos estadísticos y sistemas basados en aprendizaje automático.

## CAPÍTULO 4 - DATA MINING

---

### 4.1 Introducción

De un tiempo a esta parte, muchas de las técnicas desarrolladas en la Estadística Clásica, así como en la Inteligencia Artificial han sido puestas en práctica en un intento de construir modelos de predicción de comportamientos de forma automática y bajo una base estadística bien fundamentada.

La búsqueda de patrones útiles en datos se conoce con diferentes términos (incluyendo data mining) en diferentes comunidades (extracción de conocimiento, descubrimiento de información, procesamiento de patrones de datos,...). El término que más fuerza ha tomado es el llamado KDD (Knowledge Discovery in Databases) que se refiere al proceso no trivial de identificar patrones válidos, nuevos, potencialmente útiles, y comprensibles a partir de datos [Fay96].

En las últimas décadas, ambas comunidades científicas, la Estadística Clásica con el reconocimiento de patrones y la Inteligencia Artificial con el Aprendizaje automático han extendido sus áreas de aplicación de forma notoria, aumentando la capacidad de extraer de las grandes bases de datos, información de distintos tipos desarrollando multitud de modelos predictivos/explicativos [Hon97]. Alrededor del año 1990, ambas disciplinas aunaron esfuerzos y crearon un nuevo campo interdisciplinario que es conocido en la comunidad internacional como Data Mining (Minería de Datos). *Data mining es la aplicación de algoritmos específicos para la extracción de patrones (modelos) de los datos.* El data mining es un paso particular del proceso de KDD.

La mayoría de los algoritmos de data mining se pueden ver como una combinación de unas pocas técnicas y principios. En particular, los algoritmos de data mining consisten mayormente en una mezcla específica de tres componentes:

- *El modelo.* El componente principal. Tiene dos factores relevantes: su función (como la de clasificar, agrupar, resumir,...), y el modo de representar el conocimiento (como una función lineal de múltiples variables, en modo de árbol, de reglas, una red,...). Un modelo contiene ciertos parámetros que deben determinarse a partir de los datos.

- *El criterio de preferencia.* Es la base para escoger un modelo o un conjunto de parámetros sobre otros. El criterio suele ser la función que hace que el modelo se ajuste más apropiadamente a los datos que se disponen.
- *El algoritmo de búsqueda.* La especificación de un algoritmo para obtener modelos particulares y parámetros, los datos, el modelo (o familia de modelos), y un criterio de preferencia.

## 4.2 Funciones de los Modelos

Las funciones de los modelos más comunes en data mining incluyen:

*Clasificación:* Un clasificador es una función que asigna a una muestra no etiquetada una etiqueta o clase. Todos los clasificadores poseen una estructura de datos interna para realizar la asignación de una etiqueta a un ejemplo. Se clasifica un caso entre varias clases o categorías predefinidas. Los modelos de clasificación se pueden construir utilizando una gran variedad de algoritmos. Henery [Hen94] cataloga los algoritmos de clasificación en tres tipos:

- Extensiones de discriminación lineal (como perceptrón multicapa, discriminación lógica)
- Árboles de decisión y métodos basados en reglas (como C4.5, AQ, CART), y
- Estimadores de densidad (Naïve Bayes, k-nearest neighbor, LVQ).

*Regresión:* clasifica un caso a una variable de predicción de valor-real. En la regresión se persigue la obtención de un modelo que permita predecir el valor numérico de alguna variable.

*Clustering (agrupamiento):* clasifica un caso en una de las clases o agrupaciones en las que las clases se deben determinar a partir de los propios datos. Los clusters se definen buscando agrupaciones naturales de los datos basado en modelos de medidas de similitud, densidad de probabilidad o distancia.

*Summarization (resumen):* provee una descripción compacta de un subconjunto de datos de entrada (media y desviación estándar para todos los campos, o reglas de resumen, técnicas de visualización multivariadas, relaciones funcionales entre variables).

*Modelado de dependencias*: describe las dependencias significantes entre variables. Existen modelos de dependencias a dos niveles: el estructurado y el cuantitativo. El modelo estructurado de dependencias especifica (a menudo en modo gráfico) qué variables son localmente dependientes; el modelo cuantitativo especifica la fortaleza de las dependencias usando una escala numérica. El análisis de relaciones (como reglas de asociación), que determina relaciones existentes entre elementos de una base de datos, podría considerarse un caso particular de modelado de dependencias.

*Link analysis (análisis de conexiones)*: determina las relaciones o vínculos entre campos de la base de datos. El objetivo es el de deducir correlaciones entre campos satisfaciendo el umbral de confianza.

*Análisis de secuencias*: modela patrones secuenciales (como datos con dependencia del tiempo). El objetivo es modelar los estados del proceso generando la secuencia, o extraer y describir desviaciones y tendencias sobre el tiempo.

En el caso de la detección de intrusiones, las funciones más utilizadas son la clasificación (de un caso, en intrusión o no intrusión; o clasificar entre tipos de intrusión), clustering (los casos lejanos a las agrupaciones naturales se consideran anomalías), modelado de dependencias, o análisis de secuencias.

### 4.3 Aprendizaje Automático: Aprendizaje Supervisado

Una característica principal dentro del data mining es el paradigma de aprendizaje de los sistemas. Existen diversas definiciones de aprendizaje automático entre ellas: *“Aprendizaje denota cambios en el sistema que son adaptativos en el sentido de que permiten al sistema realizar la misma tarea, o una tomada de la misma población, la próxima vez de una forma más eficiente y efectiva.”* [Sim83]. Según Carbonnell [Car89] *“Se puede definir operacionalmente como la habilidad para realizar nuevas tareas que no podrá realizar anteriormente o realizar anteriores tareas mejor más rápidas, más exactas, etc. Como resultado de los cambios producidos por el proceso de aprendizaje”*. [For89] especificó *“El aprendizaje es un fenómeno que se muestra cuando un sistema mejora su rendimiento en una determinada tarea sin necesidad de ser reprogramado.”* En 1991 Weiss y Kulikowski [Wei91] lo explicaron como *“Un sistema que aprende es un programa de computador que toma decisiones en base a la experiencia acumulada contenida en casos resueltos satisfactoriamente. A*

*diferencia de los sistemas expertos, que resuelven los problemas utilizando un modelo por computador del razonamiento del experto humano, un sistema de aprendizaje puro puede utilizar muchas técnicas diferentes para explotar el potencial computacional del computador, sin importar su relación con el proceso cognitivo humano." Según Anzai, [Anz92]: "Cuando un sistema genera automáticamente una nueva estructura de datos o programa a partir de una existente y de esta forma irreversible cambia con algún propósito por un determinado tiempo, es lo que llamamos aprendizaje automático" Langley, 1996 [Lan96]: "Aprendizaje es la mejora en el rendimiento en ciertos entornos por medio de la adquisición de conocimiento como resultado de la experiencia en dicho entorno". Por último Mitchell [Mit97] como "Un programa de ordenador se dice que aprende de la experiencia  $E$  con respecto a una clase de tareas  $T$  y a la medida de rendimiento  $P$ , si su rendimiento en las tareas que pertenecen a  $T$  medido según  $P$ , se incrementa con la experiencia  $E$ "*

Aunque el despegue del Aprendizaje Automático se produce en los años ochenta, de ahí que las primeras definiciones mostradas anteriormente daten de esos años, la búsqueda de sistemas con capacidad de aprender se remonta a los primeros días de los computadores.

La adquisición del conocimiento por parte de los sistemas de Aprendizaje Automático se puede realizar de diferentes formas, igual que ocurre en los humanos que no tienen una única forma de aprender, aunque todos los paradigmas de aprendizaje se pueden encuadrar en las definiciones antes enunciadas, ya que todos tienen como objetivo común el incremento del rendimiento del sistema que adquiere el conocimiento.

Dentro del paradigma automático, nos encontramos con el aprendizaje supervisado, el cual genera hipótesis utilizando ejemplos con etiqueta (clase) conocida; a su vez dichas hipótesis servirán para hacer predicciones ante nuevos ejemplos con etiqueta desconocida [Bou04]. Dentro de un marco más operativo, el objetivo del aprendizaje supervisado (tanto binario como multiclase) es dividir el espacio de instancias (ejemplos) en regiones en donde la mayoría de los casos estén etiquetados con la misma clase; dicha partición es la que servirá para predecir la clase de nuevos ejemplos. Al sistema se le proporciona un conjunto de hechos etiquetados y el sistema debe obtener el conjunto de reglas que expliquen estos hechos.

## 4.4 Vecino más Cercano

El método del vecino más cercano y sus variantes están basados en la idea intuitiva de que objetos parecidos pertenecen a la misma clase, de manera que la clase a la que pertenece un objeto puede ser inferida a partir de la clase a la que pertenecen los objetos (o el objeto) de la muestra de aprendizaje que más se le parecen. La idea de parecido es reflejada formalmente en el concepto de distancia.

Los fundamentos de la clasificación por vecindad fueron establecidos por [Fix51] [Fix52] a principio de los años 50. Sin embargo, no fue hasta 1967 cuando [Cov67] enuncian formalmente la regla del vecino más cercano y la desarrollan como herramienta de clasificación de patrones. Desde entonces, este algoritmo se ha convertido en uno de los métodos de clasificación más usados [Cos93].

En este clasificador no se asume ninguna función implícita sino que la clasificación se realiza para cada nueva muestra, mediante la asignación de dicha muestra a la clase que es mayoritaria en las  $k$  muestras más próximas del conjunto de aprendizaje, siendo el caso más sencillo cuando se asigna a la clase de la muestra más cercana. Una familia de clasificadores basados en este modelo son los IB propuestos por [Aha91].

Un elemento importante que define el resultado de este tipo de métodos es la función de distancia así, un ejemplo es etiquetado con la clase de su vecino más cercano según la métrica definida por la distancia  $d$ . Normalmente se utiliza la distancia euclídea para el caso de los atributos continuos o la distancia de Hamming si se trata de atributos nominales, aunque se han propuesto otro tipo de distancias:

*Euclídea*

$$d_e(e_1, e_2) = \sqrt{\sum_{i=1}^n (e_1^i - e_2^i)^2}$$

*Manhattan*

$$d_m(e_1, e_2) = \sum_{i=1}^n |e_1^i - e_2^i|$$

Básicamente el algoritmo actúa de la siguiente manera, dado un ejemplo

Si tenemos  $m$  instancias  $\{e_1, \dots, e_m\}$  en nuestra base de datos, entonces para clasificar un nuevo ejemplo  $e'$  :

1.  $c_{min} = clase(e_1)$
2.  $d_{min} = d(e_1, e')$
3. para  $i=2$  hasta  $m$  hacer
  - $d = d(e_i, e)$
  - si  $(d < d_{min})$
  - entonces  $c_{min} = clase(e_i), d_{min} = d$
4. Devolver  $c_{min}$  como clasificación de  $e'$

donde  $d(\cdot, \cdot)$  es una función de distancia.

La regla  $NN$  puede generalizarse calculando los  $k$  vecinos más cercanos y asignando la clase mayoritaria entre esos vecinos. Tal generalización se denomina  $k$ - $NN$ . Este algoritmo necesita la especificación a priori de  $k$ , que determina el número de vecinos que se tendrán en cuenta para la predicción. Al igual que la métrica, la selección de un  $k$  adecuado es un aspecto determinante.

El algoritmo  $k$ - $NN$  se engloba dentro de las denominadas técnicas de aprendizaje perezoso (*lazy learning*), ya que no genera una estructura de conocimiento que modele la información inherente del conjunto de entrenamiento, sino que el propio conjunto de datos representa el modelo, es decir no se construye ningún modelo, el modelo es la propia base de datos o conjunto de entrenamiento. Cada vez que se necesita clasificar un nuevo ejemplo, el algoritmo recorre el conjunto de entrenamiento para obtener los  $k$  vecinos y predecir su clase. El parámetro  $k$  es un parámetro clave puesto que si se elige un  $k$  muy bajo, el sistema es sensible al ruido, en cambio si se elige un  $k$  muy alto las zonas densas pueden acaparar a las menos densas. En general es robusto al ruido cuando se usan valores de  $k$  moderados ( $k > 1$ ). Su complejidad temporal (para evaluar un ejemplo) es  $O(dn^2)$  siendo  $O(d)$  la complejidad de la distancia usada.

#### 4.4.1 Vecino más cercano e IDS

Lane realizó un IDS basado en host para la detección de anomalías basándose en IBL [Lane99]. Tomó como entradas comandos del shell de UNIX con el fin de mapear datos temporales sobre el espacio, y basó la medida de similaridad en la regla de clasificación 1-NN.

Ejemplos de utilización del clasificador k-NN son Portnoy y Eskin (del grupo de Stolfo en Columbia) [Port01] [Esk02] o Chan et al. [Cha03], que se basa en los dos anteriores. Chan realiza un trabajo comparativo entre k-NN, SVM (Support Vector Machines) y un algoritmo de clustering basado también en distancia. Yeung y Chow utilizan una estimación para la función de densidad, basándose en la ventana de Parzen [Yeu02] para la detección de tráfico anómalo. Apuntan que su estimación es muy parecida al método de k-NN. Ertöz y Steinbach utilizaron la técnica del vecino más cercano compartido (SNN: Shared Nearest Neighbour) que es particularmente apropiado para encontrar clusters o agrupaciones de diferentes tamaños, densidades y formas en los datos, principalmente en datos con gran cantidad de ruido [Ert03b].

#### 4.5 Clasificadores Bayesianos

Los clasificadores Bayesianos están basados en la formulación del Teorema de Bayes en 1763 [Bay63].

$$P(A|B) = \frac{p(A)p(B|A)}{p(B)}$$

donde:

- $p(A)$  es conocido como la probabilidad *a priori* de que el suceso  $A$  sea cierto.
- $p(A|B)$  es conocido como la probabilidad *a posteriori*, o, la probabilidad de que el suceso  $A$  sea cierto tras considerar  $B$ .
- $p(B|A)$  es conocido como verosimilitud o *likelihood*, e indica la probabilidad de que el suceso  $B$  sea cierto, asumiendo que  $A$  lo es.
- $p(B)$  es la probabilidad *a priori* de que el suceso  $B$  sea cierto. Actúa de coeficiente normalizador o estandarizador en la fracción.



Este teorema no sólo puede ser aplicado a sucesos, sino también a variables aleatorias, tanto unidimensionales como multidimensionales. Su formulación general es:

$$p(Y = y | X = x) = \frac{p(Y=y)p(X=x|Y=y)}{\sum_{i=1}^r p(Y=y_i)(p(X=x | Y=y_i))}$$

Aplicado al problema de clasificación supervisada, tenemos que  $Y = C$  es una variable unidimensional; mientras que  $X = (X_1, X_2, \dots, X_n)$  es una variable n-dimensional. “X” será la variable predictora, e “Y” la variable a predecir (la clase predicha por el modelo).

Asumiendo una función de error 0/1, un clasificador Bayesiano  $\gamma(x)$  asigna la clase con mayor probabilidad a posteriori dada una determinada instancia, es decir,

$$\gamma(x) = \underset{c}{\operatorname{argmax}} p(c | x_1, x_2, \dots, x_n)$$

donde  $c$  representa la variable clase, y  $x_1, x_2, \dots, x_n$  son los valores de las variables predictoras. Podemos expresar la probabilidad a posteriori de la clase de la siguiente manera:

$$p(c | x_1, x_2, \dots, x_n) \propto p(c) p(x_1, x_2, \dots, x_n | c)$$

Asumiendo diferentes factorizaciones para  $p(x_1, x_2, \dots, x_n | c)$  se puede obtener una jerarquía de modelos de creciente complejidad dentro de los clasificadores Bayesianos, hasta ordenes exponenciales de  $2^{m \times n}$  siendo  $m$  y  $n$  el número de dimensiones de las dos variables aleatorias.

#### 4.5.1 Naive Bayes

Naive Bayes es una técnica de clasificación descriptiva y predictiva basada en la teoría de la probabilidad del análisis de T. Bayes. Esta teoría supone un tamaño de la muestra asintóticamente infinito e independencia estadística entre variables independientes, refiriéndose en nuestro caso a los atributos, no a la clase. Con estas condiciones, se puede calcular las distribuciones de probabilidad de cada clase para establecer la relación entre los atributos (variables independientes) y la clase (variable dependiente). Concretamente, dado el ejemplo  $x = (x_1, \dots, x_n)$ , donde  $x_i$  es el valor

observado para el  $i$ -ésimo atributo, la probabilidad a posteriori de que ocurra la clase  $y_m$  teniendo  $k$  valores posibles ( $y_1, \dots, y_k$ ), viene dada por la regla de Bayes:

$$P(y_m | x_1, \dots, x_n) = \frac{P(y_m) \prod_{i=1}^n P(x_i | y_i)}{P(x_1, \dots, x_n)} \quad (1.1)$$

donde  $P(y_m)$  es la proporción de la clase  $y_m$  en el conjunto de datos; e igualmente,  $P(x_i | y_i)$  se estima a partir de la proporción de ejemplos con valor  $x_i$  cuya clase es  $y_m$ . Como podemos deducir, el cálculo de  $P(x_i | y_i)$  obliga a que los valores  $x_i$  sean discretos, por lo que si existe algún atributo continuo, éste debe ser discretizado previamente. Aplicando (1.1), la clasificación de un nuevo ejemplo  $x$  se lleva a cabo calculando las probabilidades condicionadas de cada clase y escogiendo aquella con mayor probabilidad. Formalmente, si  $Y = (y_1, \dots, y_k)$  es el conjunto de clases existentes, el ejemplo  $e$  será clasificado con aquella clase  $y_m$  que satisface la expresión:

$$\forall j \neq i \quad P(y_i | x_1, \dots, x_n) > P(y_j | x_1, \dots, x_n)$$

El clasificador bayesiano es un método sencillo y rápido. Sin embargo, para estimar el término  $P(y_m | x_1, \dots, x_n)$  es decir, las veces en que para cada categoría aparecen los valores del ejemplo  $x$ , se debe recorrer todo el conjunto de entrenamiento. Este cálculo resulta impracticable para un número suficientemente grande de ejemplos por lo que se hace necesario simplificar la expresión. Para ello se recurre a la hipótesis de independencia condicional con el objeto de poder factorizar la probabilidad.

La suposición de independencia estadística de las variables es una limitación importante, ya que este hecho es relativamente infrecuente.

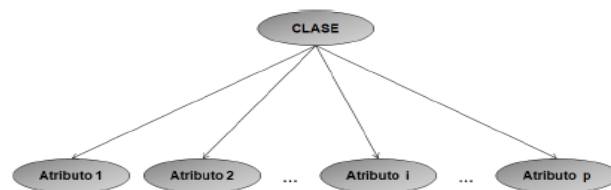


Figura 6 Estructura de un Clasificador Naive Bayes

#### 4.5.2 Redes Bayesianas

Una red bayesiana es un grafo acíclico dirigido y anotado que describe la distribución de probabilidad conjunta que gobierna un conjunto de variables aleatorias.

Sea  $X = \{X_1, X_2, \dots, X_n\}$  un conjunto de variables aleatorias.

Formalmente, una red Bayesiana para  $X$  es un par  $B = \langle G, T \rangle$  en el que:

- $G$  es un grafo acíclico dirigido en el que cada nodo representa una de las variables  $X_1, X_2, \dots, X_n$ , y cada arco representa relaciones de dependencia directas entre las variables. La dirección de los arcos indica que la variable ‘apuntada’ por el arco depende de la variable situada en su origen.
- $T$  es un conjunto de parámetros que cuantifica la red. Contiene las probabilidades  $P_B(x_i / \pi_{xi})$ , para cada posible valor  $x_i$  de cada variable  $X_i$  y cada posible valor  $\pi_{xi}$  de  $\Pi_{xi}$ , donde éste último denota al conjunto de padres de  $X_i$  en  $G$ . Así, una red bayesiana  $B$  define una distribución de probabilidad conjunta única sobre  $X$  dada por [Fri97]

$$P_B(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | \Pi_{xi})$$

o lo que es lo mismo la distribución conjunta de los valores del nodo puede ser escrita como el producto de las distribuciones locales de cada nodo y sus padres. Si el nodo  $X_i$  no tiene padres, su distribución local de probabilidad se toma como incondicional, en otro caso se considera condicional.

La topología o estructura de la red no sólo proporciona información sobre las dependencias probabilísticas entre las variables, sino también sobre las independencias condicionales de una variable o conjunto de ellas dada otra u otras variables. Cada variable es independiente de las variables que no son descendientes suyas en el grafo, dado el estado de sus variables padre. La inclusión de las relaciones de independencia en la propia estructura del grafo hace de las redes bayesianas una buena herramienta para representar conocimiento de forma compacta – se reduce el número de parámetros necesarios-. Además, proporcionan métodos flexibles de razonamiento basados en la propagación de las probabilidades a lo largo de la red de acuerdo con las leyes de la teoría de la probabilidad.

Para utilizar una red bayesiana como clasificador, un algoritmo de búsqueda determinado encuentra una red  $B$ ,  $P_B(A_1, A_2, \dots, A_n; C)$ , que mejor ajusta a un conjunto de entrenamiento  $D$  de acuerdo a alguna función de evaluación [Fri97] [Coo92]. Una vez que se determina la red,  $B$  selecciona la etiqueta  $c$  que maximiza la probabilidad posterior  $P_B(c / a_1, \dots, a_n)$  [Fri97] [Coo92].

#### 4.5.2.1 Tree Augmented Naive Bayes “TAN”

El modelo Naïve Bayes no es capaz de tratar con dependencias entre las variables predictoras. En dominios donde la independencia condicional entre las variables predictoras dada la clase no se cumple, el rendimiento de un modelo naïve Bayes puede limitarse en gran manera. El modelo bayesiano TAN o tree augmented naïve Bayes [Fri97], construye un clasificador donde existe una estructura de dependencias arborescente entre las variables predictoras. Basándose en un modelo de dependencias semejante al del naïve Bayes, añade dependencias condicionales entre los nodos, formando un árbol entre ellas. La mezcla de ambas estrategias hace posible la relajación de independencia entre las variables predictoras, en la Fig7. se muestra un ejemplo de modelo TAN.

Este modelo, propuesto por Friedman [Fri97], está basado en el cálculo de la información mutua condicionada entre pares de variables,

$$I(X_i, X_j | C) = \sum_c \sum_{x_i} \sum_{x_j} p(x_i, x_j, c) \log \frac{p(x_i, x_j | c)}{p(x_i | c) p(x_j | c)}$$

y fuerza a construir una estructura conexa de árbol con todas las variables del dominio del problema.

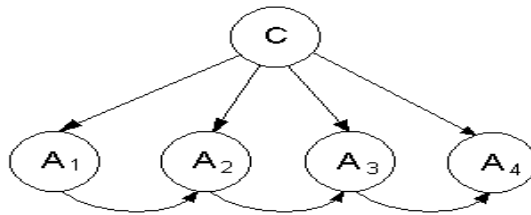


Figura 7 Estructura TAN

#### 4.5.3 Clasificadores Bayesianos e IDS

Naive Bayes es un tipo simple de redes Bayesianas particularmente eficientes en tareas de inferencia.

Axelsson publicó un artículo que utilizaba la regla Bayesiana de probabilidad condicional para apuntar las implicaciones de la falsedad de la tasa básica (base-rate fallacy) en detección de intrusos [Axe99b]. El grupo de investigación del SRI desarrolló un módulo para su IDS EMERALD llamado eBayes TCP que utilizaba tecnología de redes Bayesianas para analizar, lo que ellos llaman, “explosiones de

tráfico”. Las categorías de ataques se representan como hipótesis de modelos, las cuales se van reforzando de forma adaptativa [Val00]. Más adelante, en [Val01] presentaron un trabajo en el que también incorporaban elementos de la inferencia de Bayes, pero para la realización de correlación de alertas.

Daniel Barbará et al. también hacen uso de la teoría Bayesiana para su sistema ADAM (Audit. Data Analysis and Mining) [Bar01]. En dicho trabajo, proponen el uso de estimadores pseudo-Bayes para afinar la capacidad detectar anomalías, reduciendo a su vez el número de falsas alarmas. ADAM hace uso de reglas de asociación para la detección de anomalías, pero dicho clasificador reconoce únicamente los ataques que aparecen en el conjunto de datos de entrenamiento. Con la técnica pseudo-Bayes, no es necesario el conocimiento previo sobre los ataques, ya que las probabilidades anteriores y posteriores estimadas de los ataques nuevos se derivan de la información de las instancias normales y los ataques conocidos. Tras aplicar la técnica de pseudo-Bayes, construyen un clasificador naive Bayes para clasificar los ejemplos en normales, ataques conocidos y ataques nuevos. Sebyala también utiliza una red naive Bayes para realizar detección de intrusos sobre eventos de red [Seby02].

En [Put02] se utilizan técnicas Bayesianas para obtener parámetros con probabilidades máximas a posteriori para IDS basados en anomalías. Goldman presentó un modelo que simulaba un atacante de forma inteligente usando técnicas Bayesianas para así crear un plan de acciones en base a objetivos [Gol02].

En 2003, Kruegel et al. utilizaron redes Bayesianas para clasificar eventos basados en las salidas de los diferentes modelos utilizados para la detección de anomalías y en información adicional extraída del mismo entorno [Kru03]. El trabajo publicado por Ben Amor compara el rendimiento de redes naive Bayes con los árboles de decisión y, más concretamente, con los resultados que se obtuvieron en el campeonato KDD 99 [Amo04].

## 4.6 Árboles de Decisión

Los métodos de aprendizaje supervisado basados en árboles de decisión son uno de los métodos más populares dentro del área de la Inteligencia Artificial para tratar el problema de la clasificación [Qui86]. Un árbol de clasificación está formado por nodos, ramas y hojas. Cada nodo representa una decisión sobre los valores de un atributo

concreto. El primer nodo del árbol es conocido como el nodo raíz. Finalmente están los nodos terminales u hojas en los que se toma una decisión acerca de la clase a asignar. Así, a la hora de clasificar un nuevo caso, tendrán que compararse los valores de los atributos con las decisiones que se toman en los nodos, siguiendo la rama que coincida con dichos valores en cada test o decisión. Finalmente se llega a un nodo terminal u hoja que predice la clase para el caso tratado. Un árbol de decisión también se puede ver como un conjunto de reglas si-entonces.

Dentro de los sistemas basados en árboles de decisión, habitualmente denominados *TDIDT* (*Top Down Induction of Decision Trees*), se pueden destacar dos familias o grupos: la familia *ID3*, cuyos máximos representantes son el propio algoritmo *ID3* propuesto por [Qui86] y el sistema CLS de [Hun66]; y la familia de árboles de regresión, cuyo exponente más significativo es *Cart*, desarrollado por [Bre84].

Los *TDIDT* se caracterizan por utilizar una estrategia de divide y vencerás descendente, es decir, partiendo de los descriptores hacia los ejemplos, dividen el conjunto de datos en subconjuntos siguiendo un determinado criterio de división. A medida que el algoritmo avanza, el árbol crece y los subconjuntos de ejemplos son menos numerosos. *ID3* puede considerarse como una versión preliminar de *C4.5*, el cual resuelve algunos inconvenientes de su antecesor sobre el uso de atributos continuos, el tratamiento de valores ausentes y el proceso de poda.

#### 4.6.1 Algoritmo ID3

Cada nodo interno del árbol contiene una decisión sobre uno de los atributos, de cuyo valor dependerá el camino a seguir para clasificar un ejemplo, y cada hoja contiene una etiqueta de clase. Así, la clasificación de un ejemplo se lleva a cabo recorriendo el árbol desde la raíz hasta una de las hojas que determinará la clase del mismo. Inicialmente, el algoritmo toma todo el conjunto de datos  $D$ . Si todos los ejemplos pertenecen a una misma clase, el proceso finaliza, insertando un nodo hoja con dicha clase. En caso contrario, se selecciona aquel atributo  $A_i$  que mejor divide el conjunto de datos y se inserta un nodo con dicho atributo para establecer una decisión. Una vez creado el nodo, para cada valor distinto  $A_{iv}$  del atributo  $A_i$ , se traza un arco y se invoca recursivamente al algoritmo para generar el subárbol que clasifica los ejemplos de  $D$  que cumplen que  $A_i = A_{iv}$ . Esta llamada es realizada sin tener en cuenta el atributo

$A_i$  y substrayendo del conjunto de datos  $D$  todos aquellos ejemplos donde  $A_i \neq A_{iv}$ . El proceso se detiene cuando todas las instancias de un conjunto pertenecen a la misma clase.

El algoritmo ID3, para elegir la raíz del árbol y los posteriores atributos-nodos donde se toma una decisión, utiliza la ganancia de información. La ganancia de la información es simplemente la reducción esperada en la entropía causada al particionar los ejemplos de acuerdo a un atributo. Así, el atributo  $A_i$  seleccionado para determinar la división será aquel que mayor ganancia obtenga respecto al conjunto  $D$ ,

$$\text{Ganancia}(D, A_i) = \text{Ent}(D) - \sum_{v=1}^{|A_i|} \frac{|D(A_{iv})|}{|D|} \times \text{Ent}(D(A_{iv}))$$

donde  $|A_i|$  es el número de valores distintos de del atributo  $A_i$ .  $E(A_{iv})$  es el subconjunto de  $D$  para el cual  $A_i = A_{iv}$ , siendo  $|D(A_{iv})|$  su cardinal;  $|D|$  es el número total de ejemplos; y  $\text{Ent}(\cdot)$  es la entropía.

La entropía puede ser considerada como la *cantidad de información* contenida en el resultado de un experimento. Dicha información, dependerá sobre el *conocimiento previo* que se tiene sobre los resultados del experimento. Cuanto menos se conoce más información se obtiene (o más se aprende). Si un experimento puede tener  $m$  resultados distintos  $v_1, \dots, v_m$  que pueden ocurrir con probabilidades  $P(v_1), \dots, P(v_m)$ , la cantidad de información  $I$  que se obtiene al conocer el resultado real del experimento es:

$$I(P(v_1), \dots, P(v_m)) \equiv \sum_{i=1}^m -P(v_i) \log_2 P(v_i)$$

Para ejemplificar esta idea, consideremos el experimento de arrojar una moneda, el cual tiene como resultados posible cara y cruz. Si conocemos de antemano que la moneda fue alterada para que siempre caiga cara, la entropía (información) “ $I$ ” del resultado del experimento será:

$$I(P(\text{cara}), P(\text{cruz})) = I(1, 0) = -1 \log_2 1 - 0 \log_2 0 = 0$$

Este resultado significa que, dado que ya sabemos que la moneda caerá cara, la información que obtengamos al conocer el resultado del experimento será nula. Si en cambio utilizamos una moneda totalmente balanceada, que produce cualquiera de los dos resultados en forma equiprobable, tendremos que:

$$I\left(\frac{1}{2}, \frac{1}{2}\right) = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} = 1$$

Como podemos observar la entropía tiene su valor más bajo (0) cuando existe total certeza en el resultado del experimento, mientras que el mayor valor de entropía es alcanzado en el caso de mayor incertidumbre (eventos equiprobables).

Pseudocódigo

### *ID3 (Instancias)*

*Si todas las instancias son de la misma clase C ENTONCES  
devolver Hoja(C)*

*SINO SI el conjunto de instancias está vacío ENTONCES  
devolver Hoja (Clase\_por\_defecto)*

*SINO SI el conjunto de instancias no contiene ningún atributo  
ENTONCES devolver Hoja (Clase\_mayoritaria)*

*SINO*

*Elegir atributo A con mayor ganancia de información*

*Crear nodo con el atributo seleccionado*

*Para cada valor V del atributo A Crear una rama con el valor V*

*Seleccionar las instancias con el valor V del atributo A*

*Eliminar el atributo A de este conjunto de instancias Cv*

*Asigna a la rama el árbol devuelto por ID3(Cv)*

*Devolver nodo*

Pese a su simplicidad y bajo coste computacional, *ID3* presenta inconvenientes importantes, algunos de los cuales son corregidos por su sucesor *C4.5*. Los más evidentes son la incapacidad para trabajar con atributos continuos y tratar valores ausentes. Sin embargo, presenta una serie de problemas que afectan directamente a la precisión del árbol generado. En primer lugar, la heurística usada para establecer los test es propensa a seleccionar aquellos atributos con mayor número de valores distintos, ya que a mayor número de particiones, la entropía de cada subconjunto tiende a ser menor. En segundo lugar, *ID3* resulta muy vulnerable a la presencia de ruido e inconsistencia en los datos, lo cual ocasiona la generación de *hojas muertas* que clasifican ejemplos de más de una clase.

### 4.6.2 Algoritmo C4.5

El algoritmo C4.5 fue desarrollado por Quinlan en 1993 [Qui93], como una extensión (mejora) del algoritmo ID3 que desarrolló en 1986. Este algoritmo introduce las siguientes mejoras:

- Permite trabajar con valores continuos para los atributos, separando los posibles resultados en 2 ramas  $A_i \leq N$  y  $A_i > N$ .
- Los árboles son menos frondosos, ya que cada hoja cubre una distribución de clases no una clase en particular.
- Utiliza el método "divide y vencerás" para generar el árbol de decisión inicial a partir de un conjunto de datos de entrenamiento.



- Se basa en la utilización del criterio de proporción de ganancia (gain ratio), de esta manera se consigue evitar que las variables con mayor número de posibles valores salgan beneficiadas en la selección.

*C4.5* produce un árbol de decisión similar al de *ID3*, con la salvedad de que puede incluir condiciones sobre atributos continuos. Así, los nodos internos pueden contener dos tipos de decisión según el dominio del atributo seleccionado para la partición. Si el atributo  $A_i$  es discreto, la representación es similar a la de *ID3*, presentando una decisión con una condición de salida (rama  $A_i = A_{iv}$ ) por cada valor  $A_{iv}$  diferente del atributo.

Por contra, si el atributo  $A_i$  es continuo, el test presenta dos únicas salidas,  $A_i \leq N$  y  $A_i > N$  que comparan el valor de  $A_i$  con el umbral  $N$ . Para calcular  $N$ , se aplica un método similar al usado en [Bre84], el cual ordena el conjunto de  $t$  valores distintos del atributo  $A_i$  presentes en el conjunto de entrenamiento, obteniendo el conjunto de valores  $\{a_{i1}, a_{i2}, \dots, a_{it}\}$ .

Cada par de valores consecutivos aporta un posible umbral:

$$N = \frac{a_{iv} + a_{i(v+1)}}{2}$$

teniendo en total  $t-1$  umbrales, donde  $t$  es como mucho igual al número de ejemplos. Una vez calculados los umbrales, *C4.5* selecciona aquel que maximiza el criterio de separación. Como se mencionó anteriormente, el criterio de maximización de la ganancia de información usado en *ID3* produce un sesgo hacia los atributos que presentan muchos valores distintos. *C4.5* resuelve este problema usando la razón de ganancia (*gain ratio*) como criterio de separación a la hora de tomar una decisión.

Esta medida tiene en cuenta tanto la ganancia de información como las probabilidades de los distintos valores del atributo. Dichas probabilidades son recogidas mediante la denominada *información de separación* (*split information*), que no es más que la entropía del conjunto de datos  $D$  respecto a los valores del atributo  $A_i$  en consideración, siendo calculada como:

$$I(D, A_i) = \sum_{v=1}^{|A_i|} \frac{|D(A_{iv})|}{|D|} \times \log_2 \left( \frac{|D(A_{iv})|}{|D|} \right)$$

donde  $|A_i|$  es el número de valores distintos del atributo  $A_i$ ,  $D(A_{iv})$  es el subconjunto de  $D$  para el cual  $A_i = A_{iv}$  siendo  $|D(A_{iv})|$  su cardinal; y  $|D|$  es el número total de ejemplos. La información de separación simboliza la información potencial que representa dividir el conjunto de datos, y es usada para compensar la menor ganancia de aquellas

decisiones con pocas salidas. Con ello la razón de ganancia es calculada como el cociente entre la ganancia de información y la información de separación. Tal cociente expresa la proporción de información útil generada por la división.

$$RazonDeGanancia(D, A_i) = \frac{GananciaDeInformacion(D, A_i)}{InformacionDeSeparacion(D, A_i)}$$

C4.5 maximiza este criterio de separación, premiando así a aquellos atributos que, aun teniendo una ganancia de información menor, disponen también de menor número de valores para llevar a cabo la clasificación. Sin embargo, si el atributo incluye pocos valores, la información de separación puede ser cercana a cero, y por tanto el cociente sería inestable. Para evitar tal situación, el criterio selecciona un valor de atributo que maximice la razón de ganancia pero obligando a que la ganancia del mismo sea al menos igual a la ganancia media de todos los atributos examinados. El coste computacional para este algoritmo teniendo un conjunto de datos con  $m$  ejemplos y  $n$  atributos, el coste medio de construcción del árbol es de  $O(mn \log_2 m)$ , mientras que la complejidad del proceso de poda es de  $O(m(\log_2 m)^2)$ .

### 4.6.3 Algoritmo CART

Los árboles de clasificación y regresión (CART), es una técnica no paramétrica basada en la generación de un modelo con estructura de árbol que permita explicar o predecir una determinada variable respuesta [Bre84], que puede ser tanto categórica (árboles de clasificación) como continua (árboles de regresión).

Trabaja igual que C4.5 excepto que el criterio que se utiliza para seleccionar la mejor división de cada grupo se basa en el índice de Gini, que es una medida de la impureza de cada nodo (subgrupo). El índice de Gini para el nodo  $m$  se define como:

$$I_m = \sum_{k=1}^K p_{m,k}(1 - p_{m,k}) = 1 - \sum_{k=1}^k (p_{m,k})^2$$

donde  $k = 1, \dots, K$  son las categorías de la variable respuesta y  $p_{m,k}$  es la proporción de elementos de la categoría  $k$  en el nodo  $m$ . Esta medida alcanza su mínimo en 0, cuando todos los elementos de un nodo pertenecen a una misma clase. Para cada nodo se escoge la variable y regla de división que minimiza la suma ponderada de los índices de Gini de los subnodos generados.

#### 4.6.4 Random Forest

Random forest, introducido por Breiman en 1999, utiliza un conjunto (o bosque) formado por muchos árboles de clasificación. Para clasificar un nuevo objeto, cada árbol en el conjunto lo toma como entrada y produce una salida, su clasificación. La decisión del conjunto de árboles se toma como la clase con mayoría de votos en el conjunto [Bre01]. En Random Forest cada árbol individual se desarrolla de una manera particular:

1. Dado un conjunto de datos de entrenamiento de cardinalidad  $N$ , toma  $N$  ejemplos aleatoriamente con repetición (un bootstrap). Este será el conjunto de entrenamiento para crear el árbol.

2. Para crear cada nodo del árbol, se utiliza únicamente una pequeña cantidad de las variables del problema. Si cada objeto tiene  $M$  variables de entrada, se determina un número  $m \ll M$  y para cada nodo del árbol se seleccionan  $m$  variables aleatoriamente. La variable más relevante de este subconjunto elegido al azar se usa en el nodo. El valor de  $m$  se mantiene constante durante la expansión del bosque.

3. Cada árbol es desarrollado hasta la mayor extensión posible. No se realiza poda. [Bre01] muestra que el error del conjunto de los árboles depende de dos factores:

1. La correlación entre dos árboles cualesquiera en el bosque. El incremento en la correlación produce un incremento en el error del bosque. La utilización de un subconjunto de variables elegidas al azar y de un bootstrap de datos (remuestreo con reposición) tiende a reducir dicha correlación.

Para cada división de un nodo, no se selecciona la mejor variable de entre todas, sino que se selecciona al azar un subconjunto de variables del tamaño especificado y se restringe la selección de la variable a este subconjunto. De esta forma se incluye una mayor variabilidad de árboles y se reduce la dependencia del resultado con las divisiones precedentes.

2. La fuerza de cada árbol individual en el bosque. Un árbol con un error bajo es un clasificador fuerte. El incremento de la fuerza de árboles individuales decrementa el error del bosque. La utilización de árboles sin poda va en este sentido.

El Random Forest, establece un ranking de la importancia de las variables en la predicción de la variable respuesta. La cuestión de la medida de importancia de las variables es un punto crucial y delicado porque la importancia de una variable está

condicionada a su interacción, posiblemente compleja, con otras variables. El Random Forest calcula dos medidas de importancia distintas.

La primera, denominada MDA (Mean Decrease Accuracy), se basa en la contribución de la variable al error de predicción, es decir, al porcentaje de mal clasificados. El error de clasificación de cada árbol se calcula a partir de la parte de la muestra que ha quedado excluida de la submuestra utilizada en la construcción del árbol, generada por remuestreo. Para calcular la importancia de cada una de las variables que aparecen en un árbol se permutan aleatoriamente los valores de esa variable, dejando intactos el resto de variables, y se vuelven a clasificar los mismos individuos según el mismo árbol pero ahora con la variable permutada. La importancia en ese árbol se calcula como el aumento en el error de predicción resultante. Finalmente se calcula la medida MDA, como la media de estos incrementos en todos los árboles en donde interviene la variable.

La segunda medida de importancia, denominada MDG (Mean Decrease Gini), se calcula a partir del índice de Gini. Éste es el criterio que se utiliza para seleccionar la variable en cada partición en la construcción de los árboles y que comporta una disminución de esta medida. La importancia de una variable en un árbol se mide como la suma de los decrementos atribuidos a esa variable y la importancia final, como la media en todos los árboles.

#### 4.6.5 Algoritmo Naive Bayes Tree

La variante *NBTREE* (*Naive-Bayes Tree*) [Koh96] presenta un algoritmo híbrido entre los árboles de clasificación y el clasificador naive Bayes. Se puede definir *NBTREE* como un árbol de clasificación cuyas hojas son clasificadores naive-Bayes como muestra la figura

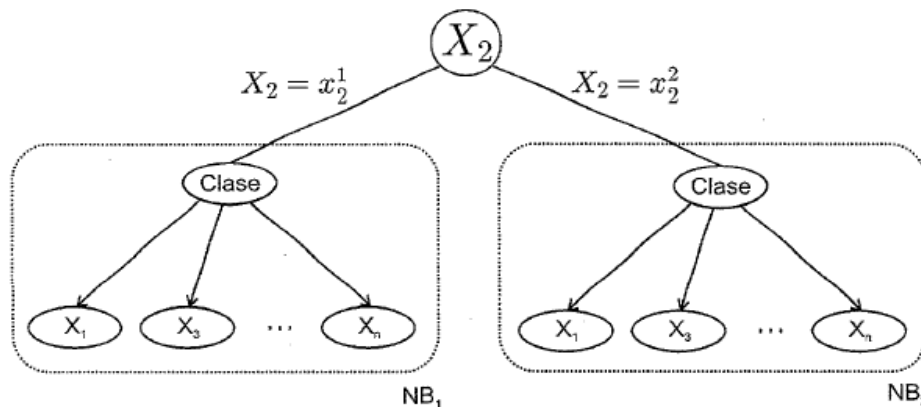


Figura 8 NBTTree con un nodo de decisión ( $X_2$ ) y 2 clasificadores NB como hojas.

Cada hoja del NBTREE contiene un clasificador naive-Bayes local que no considera las variables que se encuentran involucradas en la decisión que está en el camino que lleva hasta la hoja. Las propiedades de este árbol de clasificación son: cada nodo interno representa un atributo, cada nodo interno tiene tantos hijos o ramas salientes como valores tiene el atributo representado en dicho nodo, todas las hojas están al mismo nivel y en cualquier camino que se recorra desde la raíz hasta las hojas no existen variables repetidas. La condición “todas las hojas están al mismo nivel” se impone para simplificar el modelo, pero puede ser no tenida en cuenta en la práctica. En el trabajo [Pl102] se presenta un algoritmo heurístico para el aprendizaje de este tipo de estructuras. Este algoritmo está basado en la verosimilitud marginal de los datos para realizar la búsqueda.

#### 4.6.6 Árboles de Decisión e IDS

Nong Ye et al., de Arizona State University, realizaron una propuesta de árboles de decisión como método de aprendizaje de firmas de ataques [Ye00a] [Ye00b]. En el trabajo, utilizaron un árbol basado en el algoritmo ITI (Incremental Tree Induction) propuesto por Utgoff [Utg97]. Dicho clasificador aprendía firmas de intrusiones en su fase de entrenamiento, para después clasificar en diferentes estados las actividades de sistemas informáticos, y predecir la posibilidad de que ocurra un ataque. Más tarde, Xiangyang Li y Nong Ye realizaron otro experimento [Li01] con los algoritmos CHAID y GINI, utilizando la herramienta Answer Tree de SPSS y datos de auditoría de BSM (Basic Security Module) de Solaris. Los experimentos tienen resultados muy interesantes tanto para datos puros (limpios), como para datos con ruido. Concluyen que los algoritmos de árboles de decisión deben tener la habilidad de realizar aprendizaje incremental, pero que éstos deben ser computacionalmente asequibles, y escalables para conjuntos grandes de datos.

En el 2002, los chinos Hong Han, Xian-Liang Lu y Li-Yong Ren hacen uso del data mining para la generación de firmas de ataques de manera automática para su uso en detección de uso indebido en redes [Hong02]. Su objetivo es crear una herramienta de data mining para ayudar a expertos en el descubrimiento de firmas o patrones de ataque; herramienta que la llaman SigSniffer.

En [Kru02] se hace uso de árboles de decisión como método de optimización de IDS basados en análisis de firmas. Construyen una variante del algoritmo ID3 a partir de las firmas de Snort 2.0 y demuestran que mejora el proceso de detección. Finalmente, en [Zhi03] se aprovechan las diferentes habilidades para la clasificación que aportan tanto las redes neuronales como el algoritmo de árboles de decisión C4.5 para la detección de uso indebido.

## 4.7 Inducción de Reglas

La inducción de reglas es muy usada en problemas de clasificación. El objetivo es crear reglas a partir de un conjunto de datos. Las mismas deben recoger todo el conocimiento generalizable, sin los datos, y resultar tan pequeñas como sea posible. Además se debe garantizar que en el proceso de clasificación de un nuevo ejemplo no tenga que hacerse uso de un número elevado de reglas. Básicamente los sistemas de aprendizaje de reglas representan un paradigma transparente, fácilmente comprensible y aplicable.

En general, una regla de decisión es una regla del tipo “*Si P Entonces C*”, donde P es un predicado lógico sobre los atributos, cuya evaluación cierta, implica la clasificación con etiqueta de clase C. Desde el punto de vista de la interpretación humana, esta representación del conocimiento resulta a menudo más clara que los árboles de decisión, sobre todo en aplicaciones reales donde el número de nodos de éstos tienden a aumentar. Esto es debido tanto a la propia estructura como a las técnicas utilizadas para generar éstas. Como se vió en el apartado anterior la construcción de los árboles de decisión se basa en una estrategia de *división*, esto es, dividir el conjunto de datos en dos subconjuntos considerando un único atributo seleccionado por una heurística particular. Por el contrario, el aprendizaje de reglas sigue una estrategia de *cobertura*, esto es, encontrar condiciones de reglas teniendo en cuenta todos los atributos de forma que se cubra la mayor cantidad de ejemplos de una misma clase, y la menor del resto de las clases. Muchas de las técnicas utilizadas en los sistemas de aprendizaje de reglas fueron adaptadas del aprendizaje de árboles de decisión, el cual se basa en: la estrategia de aprendizaje conocida como *overfit-and-simplify*, y la técnica de poda (o *pruning*) conocida como REP (*reduced error pruning*).

#### 4.7.1 Algoritmo RIPPER “Repeated Incremental Pruning to Produce Error Reduction”

Poda repetida incrementada para reducir errores, es uno de los primeros algoritmos de aprendizaje de reglas simple para llevar a cabo la clasificación. RIPPER [Coh95] fue desarrollado como mejora para el algoritmo IREP. En RIPPER, una regla posee el formato análogo al de una regla de clasificación con un término antecedente y un término consecuente que posee sólo el atributo etiqueta.

SI condiciones ENTONCES acciones

Sin embargo, antes de realizar el entrenamiento el algoritmo efectúa la ordenación del conjunto de entrenamiento de manera ascendente según la frecuencia de cada clase. En este contexto, primeramente se encuentra un primer conjunto de reglas,  $CR_1$ , que posee la primera clase encontrada,  $c_1$ , de la lista ordenada de clases. En el momento que se induce una regla, se eliminan todas las tuplas que coinciden con la misma y así sucesivamente para cada regla referente a  $c_1$ . A continuación, el algoritmo repite el mismo procedimiento para las siguientes clases.

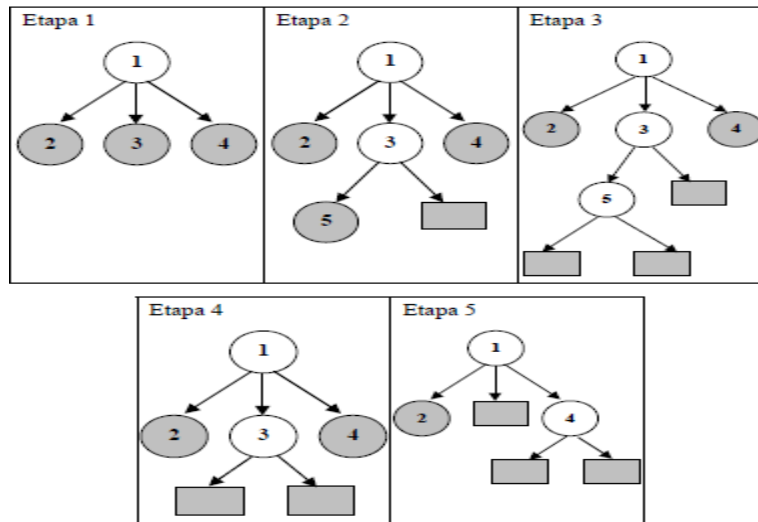
Pero pueden restar tuplas que no coincidan con ninguna regla de ninguna clase. En este caso, dichas tuplas son apartadas en un conjunto llamado creciente (*growing set*) y posteriormente en otro llamado conjunto de poda (*pruning set*). En el primer conjunto, se inducen más reglas basándose en las ya existentes, las cuales son especializadas, agregándose más ítems en los términos antecedentes. Posteriormente, en el conjunto de poda, dichas reglas son generalizadas de manera sucesiva, eliminándose ítems de los términos antecedentes de las mismas.

#### 4.7.2 Algoritmo PARTIAL Decision Tree : “PART”

El algoritmo PART de aprendizaje de reglas basado en árboles de decisión parciales [Fra98] representa un enfoque alternativo híbrido para la inducción de reglas. Básicamente construye una regla, elimina las instancias que ésta cubre y continúa creando reglas recursivamente para las instancias que permanecen hasta que no quede ninguna, pero para crear una regla, se construye un árbol de decisión podado a partir

del conjunto activo de instancias, la hoja de éste con mayor cobertura se convierte en una regla, y se desecha el árbol (recordemos que como se citó en el sub-apartado árboles de decisión, un árbol de decisión se puede ver como un conjunto de reglas si-entonces). Aunque el hecho de construir repetidamente árboles de decisión para simplemente descartar la mayoría de ellos pueda resultar un tanto extraño, en verdad resulta que el empleo de un árbol podado para obtener una regla en vez de construirla incrementalmente añadiendo conjunciones evita la tendencia a la “sobrepoda”. Construir un árbol de decisión completo para obtener una única regla supondría un enorme derroche de recursos, pero en el caso del algoritmo PART: la idea clave es construir un árbol de decisión parcial en vez de uno completo. Un árbol de decisión parcial contiene algunas ramas que representan subárboles no definidos. Para generar tal árbol parcial, se integran las operaciones de construcción y poda con el objetivo de encontrar un subárbol “estable” que no pueda simplificarse más. Una vez hallado este subárbol, la construcción del árbol cesa y dicho subárbol se convierte en una regla. Para la construcción del árbol se procede igual que en el algoritmo de construcción de árboles C4.5, se escoge un atributo-nodo para ser dividido y se evalúa su entropía, los subconjuntos resultantes se expanden en orden creciente de acuerdo con su entropía, empezando con el de menor entropía, debido a que es más probable que la expansión de los subconjuntos de baja entropía finalice rápidamente y dé lugar a subárboles de pequeño tamaño y por lo tanto a reglas más generales. La expansión se va realizando recursivamente, pero tan pronto como aparezca un nodo interno cuyos hijos ya se hayan expandido en hojas, se comprueba si dicho nodo interno puede ser sustituido por una única hoja, esto es, se intenta “podar” ese subárbol, y la decisión acerca de esta poda se toma de la misma manera que en C4.5. Si el reemplazo se lleva a cabo, se vuelve hacia atrás a explorar los nodos hermanos del nodo reemplazado. Sin embargo, si durante la exploración se encuentra un nodo cuyos hijos no sean todos hojas, los subconjuntos restantes ya no se explorarán y, por tanto, los subárboles correspondientes no serán definidos, deteniéndose automáticamente la generación del árbol. La siguiente figura muestra un ejemplo del proceso:





Fuente: Frank y Witten (1998).

Figura 9 Ejemplo Algoritmo Part

Desde la etapa 1 hasta la 3, se lleva a cabo la construcción del árbol recursivamente del modo usual, pero escogiendo para la expansión el nodo con la entropía más baja, en este ejemplo, el nodo 3 entre las etapas 1 y 3. El resto de nodos circulares todavía no son expandidos. Los nodos rectangulares representan hojas. Entre las etapas 2 y 3, el nodo rectangular tendrá una entropía más baja que su hermano, el nodo 5, pero no puede ser expandido porque ya es una hoja. Entonces se vuelve hacia atrás y el nodo 5 resulta elegido para su expansión. Cuando se alcanza la tercera etapa existe un nodo cuyos hijos son todos hojas, el nodo 5, y esto desencadena el proceso de poda. Se plantea la posibilidad de reemplazar este subárbol, y se acepta tal reemplazo, lo que conduce a la etapa 4. Ahora se considera el nodo 3 para su reemplazo, y de nuevo es aceptado. El retroceso continúa y ahora resulta que el nodo 4 tiene una entropía más baja que el 2; entonces el nodo 4 se expande en 2 hojas. Se estudia la posibilidad de su reemplazo, y supongamos que el nodo 4 resulta no ser reemplazado. En este punto, el proceso finalizaría, habiéndose obtenido el árbol parcial de 3 hojas de la etapa 5.

Una vez construido un árbol parcial, se extraerá una única regla a partir de él. Cada una de sus hojas se corresponde con una regla posible, y se escogerá la que cubra el mayor número de instancias, puesto que proporcionará la regla más general. Si se está construyendo un árbol parcial y existen instancias con valor desconocido para alguno de los atributos implicados, su tratamiento será similar al empleado en el algoritmo C4.5. Cuando la lista de decisión obtenida vaya a ser utilizada para clasificar una nueva instancia con atributos desconocidos, se generará una distribución de probabilidad sobre las clases correspondientes a las distintas reglas que le puedan ser aplicadas. La fracción

del caso que se asigna a cada una de estas reglas vendrá dada por el porcentaje de casos de entrenamiento que llegando a la regla son cubiertos por ella. Finalmente, la clase más probable de acuerdo con la distribución de probabilidad así obtenida será la que se asigne a la nueva instancia que se está clasificando. De acuerdo con los experimentos realizados por sus creadores, el algoritmo PART produce con gran rapidez conjuntos de reglas tan o más precisos que otros métodos rápidos de inducción de reglas. Pero su principal ventaja sobre otras técnicas no es el rendimiento sino la simplicidad, y ello se consigue combinando el método de inducción *top-down* de árboles de decisión con la estrategia *separate-and-conquer* de aprendizaje de reglas. Estas razones son las que nos han conducido a decantarnos por dicho algoritmo para la realización de nuestro trabajo.

### 4.7.3 Inducción de Reglas e IDS

En 1990, Teng, Chen y Lu propusieron un método para descubrir patrones secuenciales temporales en una secuencia de eventos. De este modo, con el sistema Time-based Inductive Machine (TIM), se podían aprender patrones secuenciales para detector intrusos [Teng90].

Ripper ha demostrado ser más efectivo que el algoritmo de árboles de decisión C4.5 para datos con gran cantidad de ruido. Además genera un tipo de reglas fáciles de entender y de traducir a un lenguaje como Prolog. El sistema se compone de un conjunto de reglas de asociación y patrones de episodios frecuentes que pueden ser aplicados tanto a eventos de seguridad como a conjuntos de datos de tráfico de red [Bru04]. Wenkee Lee y Sal Stolfo fueron los que propusieron el uso de RIPPER para la detección de intrusos en su proyecto JAM [Sto97] [Lee99], en el que proponen un innovador método para crear modelos de IDS y reglas de forma automática.

## 4.8 Lógica Difusa

Los sistemas de control difuso permiten describir el conjunto de reglas que utilizaría un ser humano que controlase el proceso, con toda la imprecisión que poseen los lenguajes naturales. La teoría de subconjuntos difusos relaja el concepto de pertenencia de un elemento a un conjunto. En la teoría tradicional, un elemento simplemente pertenece o no a un conjunto. Sin embargo, en la teoría de subconjuntos difusos, un elemento pertenece a un conjunto con un cierto grado de certeza. Aplicando esta idea, el

uso de la lógica difusa permite un mejor tratamiento de la información cuando ésta es incompleta, imprecisa o incierta. Por ello, ha sido aplicada por muchos autores en tareas de clasificación, usando a menudo reglas difusas como representación del conocimiento. Estos sistemas son denominados tradicionalmente *Fuzzy Rule-Based Classification Systems*. Las reglas difusas (*fuzzy rules*) presentan varias diferencias respecto a las reglas de decisión vistas en el capítulo anterior. Por un lado, las condiciones del antecedente de una regla difusa no son creadas en base a valores concretos ni rangos numéricos determinados, sino a etiquetas lingüísticas. Por ejemplo, los términos *medio*, *alto* y *bajo* son imprecisos, pero asociados a una semántica que les asigne un significado, podrían ser etiquetas lingüísticas para describir la altura de un objeto. Por otro lado, en el consecuente de la regla pueden aparecer una o varias etiquetas de clase, así como el grado de certeza o solidez asociado a cada clase en una regla concreta. Cuando se les proporciona el valor actual de las variables de entrada se obtiene el valor de las variables de salida, calculado mediante un método de inferencia difusa. Así, la estructura de una regla difusa es la siguiente:

$$R_l : \text{Si } a_1 \text{ es } A_1^k, \dots, a_m \text{ es } A_m^k \text{ Entonces } C$$

donde cada  $a_i$  es un atributo del conjunto de datos;  $A_i^k$  son las etiquetas lingüísticas para el atributo  $a_i$  en la regla  $R_l$ ; y  $C$  representa el consecuente de la regla.

Con respecto a la complejidad de la representación mediante reglas difusas, recordemos que la forma en la que el ser humano expresa sus ideas es muy similar al modo en que los sistemas difusos representan el conocimiento.

#### 4.8.1 Algoritmo Fuzzy Unordered Rule Induction Algorithm

Furia [Hüh09], Algoritmo de Inducción de Reglas de Asociación Borrosas no Ordenadas, aplica reglas difusas en el problema de clasificación.

Este algoritmo se basa, en el algoritmo RIPPER [Coh95] visto en el apartado 4.6.1. El algoritmo utiliza los mismos procedimientos de RIPPER, sin embargo, en lugar de producir reglas clásicas, produce reglas de asociación borrosas. FURIA no induce reglas de manera ordenada, pues, según los autores [Hüh09], ordenar la inducción de reglas puede comprometer la comprensibilidad, pues el término antecedente de cada regla

contiene, de manera implícita, la negación de todas reglas precedentes. De esta manera, el algoritmo induce reglas sin restringir que las mismas se refieran solamente a una determinada clase cada vez, y tampoco se utiliza una regla patrón (*default rule*) para clasificar un ejemplo que no coincida con ninguna regla. En consecuencia, FURIA aprende a separar cada clase de todas las demás clases, lo que significa que ninguna norma por defecto se utiliza y el orden de las clases es irrelevante. Asimismo, el FURIA no efectúa la etapa de poda en las reglas, pues el conjunto de reglas inicial es inducido directamente a través de todo el conjunto de entrenamiento. En lugar de la poda, se realiza, sin embargo, la generalización de reglas inducidas. Para ello, se forma una lista ordenada, para cada regla, de los ítems presentes en su término antecedente, en la que la ordenación se refleja de acuerdo a la importancia (frecuencia) de cada ítem. De esta manera, se puede generalizar las reglas de manera más eficiente, pues se suprime en primer lugar el ítem menos importante de una determinada regla y, si es necesario (i.e. hasta que haya tuplas que coincidan con la regla en cuestión), se suprimen otros ítems respetando el mismo método. Por lo tanto, se puede afirmar de manera simplista que una regla borrosa puede ser obtenida reemplazando los intervalos de las reglas clásicas por intervalos borrosos, los cuales, son establecidos por el algoritmo a través de conjuntos borrosos definidos por una función de pertenencia trapezoidal. Para cada término antecedente de cada regla se busca un intervalo borroso que posea la misma estructura del intervalo original y cuyo rango de valores se encuadre en el valor original. Por otra parte, no se considera ninguna propiedad de la lógica borrosa en el término consecuente de las reglas generadas por FURIA, pues al clasificar un ejemplo, y en el caso de que haya más de una regla que coincida con los atributos del ejemplo a clasificar, el algoritmo elige únicamente la regla que presente mayor valor de soporte. En caso de que el soporte de dos o más reglas sea igual, se considera la regla que posea la mayor frecuencia.

#### 4.8.2 Lógica Difusa e IDS

La lógica fuzzy [Zad65] resulta adecuada en el problema de la detección de intrusos por dos razones principales. Por un lado, están involucradas una gran cantidad de características cuantitativas, y por el otro, la seguridad en sí misma incluye la confusión, es un hecho borroso [Bri00]. Dada una característica cuantitativa, se puede

usar un intervalo para indicar un valor normal. Los sistemas de detección de intrusos basados en la lógica fuzzy o lógica difusa han ido tomando fuerza en los últimos años.

El primer trabajo sobre el uso de la aplicación de la lógica fuzzy en el área de la seguridad informática que se conoce es el de T.Y. Lin, de la universidad norteamericana del estado de San José [Lin94]. Sin embargo, ha sido a partir del año 2000 cuando comienzan a realizarse multitud de trabajos sobre detección de intrusos que incorporan componentes de la lógica fuzzy.

Se pueden encontrar otros trabajos aislados que relacionan la lógica fuzzy con los IDS. Zhang Jiang et al. [Zhan03] hacen uso de la teoría fuzzy por defecto para el motor de razonamiento y respuesta de los IDS. Con su experimento demuestran que su técnica aumenta la velocidad de detección y disminuye el costo acumulado de la detección de intrusos en relación a las respuestas no estáticas, en comparación con los IDS basados en sistemas expertos tradicionales. Tratando de solventar el problema que tienen actualmente los IDS en cuanto a su excesivo número de falsos positivos, en la universidad Carlos III de Madrid utilizan umbrales fuzzy para mejorar la predicción cuando se trabaja con diferentes IDS [Orf03].

Finalmente, Jian Guan et al. [Gua04] utilizan un conjunto de reglas fuzzy para definir el comportamiento normal y anómalo de una red.

## 4.9 Algoritmos Genéticos

Los algoritmos genéticos son algoritmos de búsqueda inspirados en los mecanismos de selección natural de las especies y la combinación genética que se presenta en la reproducción de los individuos. Históricamente, los algoritmos genéticos fueron la primera técnica evolutiva utilizada. Holland [Hol75] fue quien introdujo el concepto de algoritmo genético debido a que eran algoritmos que realizaban simulaciones de poblaciones de cromosomas que se codifican como cadenas de bits.

Esta técnica fue creada por John Holland y descrita por él mismo en su libro *Adaptation in Natural and Artificial Systems* [Hol92]. Estos algoritmos utilizan una estructura de datos simple llamada cromosoma para representar posibles soluciones a un problema específico, y aplica a esas estructuras diferentes operadores y combinaciones de ellos de forma que la información importante sea preservada.

Los algoritmos genéticos generalmente han sido asociados a funciones de optimización, pero el rango de problemas a los cuales han sido y pueden ser aplicados es bastante amplio [Whi93]. Los elementos básicos de un algoritmo genético son los siguientes [Nae04], [Gol05]:

- **Población:** Es un conjunto de individuos que representan posibles soluciones al problema. Estos individuos son cadenas de bits que son evaluadas después de ser decodificadas a números reales o enteros que representan las variables del problema. Generalmente la población inicial es generada en forma aleatoria. A partir de un proceso de selección natural aplicado sobre la población inicial y mediante el uso de operadores genéticos, como el cruzamiento y la mutación, se originan los descendientes que constituirán una nueva generación.
- **Gen o Cromosoma:** Conocido también como genotipo. Es un individuo o elemento de la población, que representa una posible solución al problema.
- **Función Fitness:** Conocida también como función de aptitud. Es una expresión matemática para evaluar la aptitud (calidad) de los individuos en una generación. Lo clave a la hora de definir una función de aptitud es que esta debe devolver los valores más altos cuando es aplicada a los individuos que más se aproximan a la solución óptima.
- **Selección natural de padres:** Es un mecanismo de selección aplicado sobre una población o una generación en forma probabilística de acuerdo al valor de la función de aptitud de cada individuo. Los individuos mejor calificados de acuerdo a esta función tendrán una mayor oportunidad de ser escogidos como padres para producir la siguiente generación.
- **Operadores genéticos:** Son los operadores que permiten obtener una nueva generación a partir de una población. Los operadores genéticos más comunes son el crossover (cruce o recombinación genética) y el operador de mutación.
- **Crossover:** Es el proceso mediante el cual dos individuos se aparean para producir descendencias individuales. Esto se realiza intercambiando segmentos de los cromosomas de los padres. Se han propuesto diferentes modelos de crossover como el punto simple, el punto múltiple y el cruzamiento uniforme.
- **Mutación:** Es un mecanismo necesario para asegurar la diversidad en la población. De forma aleatoria se selecciona un individuo para sufrir la mutación, el algoritmo cambia un bit también en forma aleatoria. Esto tiene como objetivo evitar un

modelo fijo de soluciones que haya sido propagado a través de todas las diferentes generaciones.

Habiendo definido los elementos, podríamos resumir el algoritmo de la siguiente forma:

Idea Básica

- Partiendo de una población inicial (soluciones factibles)
- Seleccionar individuos (favorecer a los de mayor *calidad*)
- Recombinarlos
- Introducir mutaciones en sus descendientes
- Insertarlos en la siguiente generación

```
algoritmo genético
principio
  t:=0;
  inicializa P(t);
  evalúa P(t);
  mq not termina hacer
    t:=t+1;
    P(t):=selecciona P(t-1);
    recombina P(t);
    muta P(t);
    evalúa P(t)
  fmq;
fin
```

Figura 10 Algoritmo Genético

#### 4.9.1 Algoritmos Genéticos e IDS

La utilización de algoritmos genéticos para la detección de intrusos se ha llevado a cabo principalmente con el fin de mejorar la eficiencia seleccionando subconjuntos de características para reducir el número de características observadas manteniendo, o incluso mejorando, la precisión del aprendizaje.

Los algoritmos genéticos han sido aplicados a la seguridad informática desde principios de los noventa. El francés Ludovic Mé planteó el uso de dichos algoritmos como método para analizar los rastros de auditorías de seguridad entre 1993 y 1996

[Mé93][Mé96]. Al año siguiente presentó su tesis inspirada en este mismo tema, y tras varios trabajos, en 1998 presenta el proyecto GASSATA (Genetic Algorithm as an Alternative Tool for Security Audit Trail Analysis) [Mé98] que utiliza un algoritmo genético para buscar la combinación de los ataques conocidos que mejor se correspondan con el evento (o registro de auditoría) observado.

En 1999 en la Universidad de Iowa, Helmer et al. utilizaron algoritmos genéticos como método de seleccionar subconjuntos de características a partir de vectores de características que describían las llamadas al sistema ejecutadas por procesos con privilegios [Hel99]. Dicha selección permitía reducir significativamente el número de características necesarias para la detección sin que ello afectara a la precisión.

En la universidad de Mississippi [Wei04] llevaron a cabo experimentos con algoritmos genéticos en que inicialmente se clasifica un conjunto de conexiones de red entre normal o intrusiva de forma manual. El algoritmo genético se inicia con un pequeño conjunto de reglas generadas aleatoriamente, y dichas reglas evolucionan hasta generar un conjunto de datos mayor que contiene las nuevas reglas del IDS [Wei04].

#### 4.10 Sistema Inmune Artificial

Existen diversas definiciones de sistema inmune artificial entre ellas las siguientes, *Los sistemas inmunes artificiales son metodologías para la manipulación de datos, clasificación, representación y razonamiento, los cuales siguen el paradigma biológico del sistema inmune humano* [Sta], o también *“El sistema inmune artificial es un sistema computacional basado en los principios del sistema inmune natural”* [Tim00]. Para [Das99] *“Los sistemas inmunes artificiales son metodologías inteligentes inspiradas en el sistema inmune, enfocadas a resolver problemas del mundo real”*.

##### 4.10.1 Principio de Selección Clonal

El principio o teoría de selección clonal, plantea una explicación de cómo hace el sistema inmune para describir las características básicas de una respuesta inmune a un estímulo antigénico. Este principio establece la idea de que sólo aquellas células que reconocen a los antígenos proliferan; de esta manera son seleccionadas aquellas que tienen la capacidad de reconocerlos.



En términos generales el principio de selección clonal funciona de la siguiente manera, cuando un anticuerpo reconoce al antígeno, éste es seleccionado para que prolifere y produzca anticuerpos con su misma estructura química en grandes volúmenes. La reproducción es asexual y se realiza a través de la mitosis. Por lo anterior no existe un cruce entre anticuerpos. A los “hijos idénticos” de cada anticuerpo seleccionado se les conoce como clones. Es en ellos donde se realiza la adaptación que consiste en someterlos a un proceso de mutación con altos porcentajes. Del resultado de este proceso se obtienen nuevos anticuerpos y mediante un proceso de selección se mantienen aquellos que tengan cierto grado de afinidad con respecto a los antígenos reconocidos al principio del proceso. Aquellos anticuerpos que no fueron mantenidos se desechan y regresan al torrente sanguíneo para poder reutilizar las proteínas que los formaban en la creación de nuevos individuos. Aquellos que sí fueron seleccionados se guardan como células de memoria donde se mantienen por algún tiempo para ser utilizadas, de ser necesario, en un futuro. Los procesos y la interacción entre la mutación y la selección son análogos a la selección natural de las especies. Hay dos puntos importantes desde el punto de vista computacional en este esquema

1. La proliferación de un anticuerpo es directamente proporcional a la afinidad de éste con respecto a un antígeno detectado. Entre mayor sea la afinidad entre componentes mayor sería la cantidad de descendientes y viceversa.

2. La mutación de cada uno de los clones es inversamente proporcional a la afinidad entre el anticuerpo que los produjo y el antígeno detectado. Entre mayor sea la afinidad entre componentes menor sería el porcentaje de mutación y viceversa.

De Castro y Timmis [Cas02] desarrollaron un primer algoritmo basado en el principio de selección clonal para reconocimiento de patrones.

#### 4.10.2 Algoritmo de Selección Clonal :Clonalg

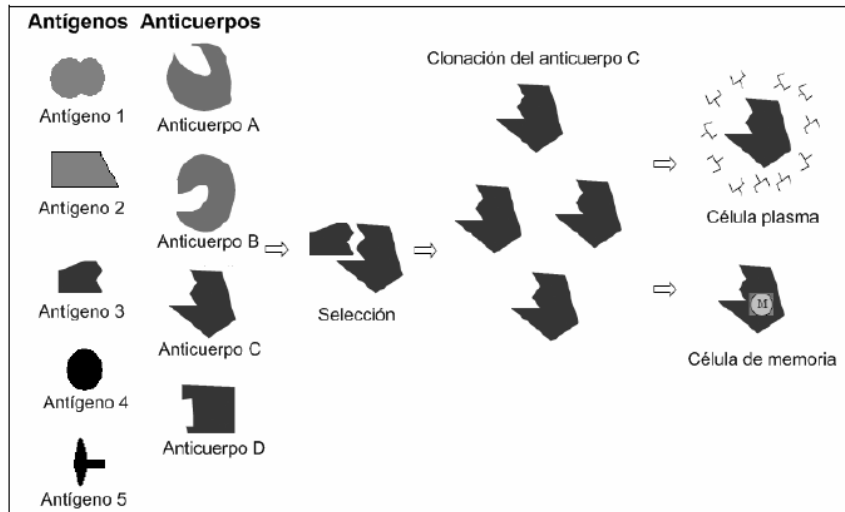


Figura 11 Selección Clonal.- Algoritmo Clonalg

Antes de pasar a explicar este algoritmo en la figura 11. se puede observar cómo trabaja un algoritmo basado en selección clonal para el reconocimiento de patrones.

Básicamente como se expuso en líneas más arriba, la idea es emular al sistema inmune natural, una vez que un antígeno es detectado por un anticuerpo, se selecciona ese anticuerpo, se clona y se almacena en una célula de plasma o de memoria. El concepto de memoria es porque estas células almacenan el modelo del antígeno para futuras infecciones. Lo mismo ocurre con un algoritmo inmune artificial, aprende a detectar patrones en la fase de entrenamiento y guarda esa información para posteriormente en la fase de test poder clasificar nuevos ataques.

El algoritmo de selección clonal (CLONALG) [Cas02], representa una implementación computacional del principio de selección clonal, es utilizado tanto para clasificación de patrones como para temas de optimización. La idea básica de Clonalg es que el algoritmo asume que cuando un anticuerpo reconoce un antígeno con un cierto grado de afinidad, entendamos afinidad como una medida de distancia, en este caso la distancia de hamming, éste tiende a proliferar y generar clones. La nueva población de clones es sometida a un proceso de mutación, proporcional a su afinidad: el clon que tiene una afinidad más alta, es clon el que tiene un porcentaje de mutación más pequeño. (la hipermutación es un operador que modifica la solución con un ratio inversamente proporcional a su *fitness*)

Pseudocódigo del algoritmo:

- 1.- *Generar aleatoriamente una población inicial  $Ab$ . Compuesta por 2 subconjuntos  $Ab_m$  (población de memoria) y  $Ab_r$  (población de reserva)*
- 2.- *Crear un conjunto de patrones de antígenos  $A_g$ .*
- 3.- *Seleccionar un Antígeno  $Ag_i$  de la población  $A_g$ .*
- 4.- *Para cada miembro de la población  $Ab$  calcular su afinidad con el antígeno  $Ag_i$  utilizando una función de afinidad  $f$  (ej. Distancia de Hamming).*
- 5.- *Seleccionar los  $n$  anticuerpos con mayor afinidad y generar un número de clones por cada anticuerpo en proporción a su afinidad, para formar una nueva población  $P$*

$$NC(ab_i) = \text{round} \left( \frac{\beta \cdot N}{i} \right), i = 1, \dots, n$$

donde  $ab_i$  es el  $i$ -ésimo mejor anticuerpo de la población actual, "round" es un operador que redondea su argumento al entero más cercano y  $\beta$  es un factor multiplicador.

- 6.- *Mutar los clones de la población  $P$  de manera inversamente proporcional a su afinidad para producir una población más madura  $P'$ .*

$$\sigma(ab) = \max \left( 1, \text{round} \left( D \cdot e^{-k \frac{f(ab)}{f^*}} \right) \right)$$

Donde  $k$  es el factor de control de la declinación y  $f^*$  es el valor de afinidad de los mejores anticuerpos de la población actual. La tasa de hipermutación indica el número de mutaciones simples que se aplican a los anticuerpos clonados. Una simple mutación consiste en elegir al azar dos posiciones de la secuencia que representa el anticuerpo y el intercambio de ellos. A fin de mantener los mejores anticuerpos, mantenemos un original (padres) de anticuerpos no-hypermutados.

- 7.- *Reaplicar la función de afinidad para cada uno de los miembros de la población  $P'$ , seleccionar la mayor afinidad como candidato de la célula de memoria. Si su afinidad es mayor que la célula de memoria actual  $Ab_{mi}$ , entonces el candidato pasa a ser la nueva célula de memoria.*
- 8.- *Borrar aquellos anticuerpos con baja afinidad de la población  $Ab_r$  y reemplazarlos con nuevos miembros generados al azar.*
- 9.- *Repetir los pasos 3-8 hasta que todos los antígenos hayan sido presentados a todos los anticuerpos. Esto representa una generación del algoritmo.*

El algoritmo tiene 5 parámetros:  $N$  tamaño de población,  $n$  número de los mejores anticuerpos para ser clonados,  $\beta$  factor multiplicador para calcular el número de clones dado un anticuerpo,  $k$  factor de control de declinación de la tasa de hipermutación y  $d$  el número de nuevos anticuerpos generados para ser añadidos a la población.

La hipermutación (mutación) contribuye a la introducción de diversidad en los anticuerpos seleccionados, permitiendo la adaptación rápida de la respuesta inmune. Pero por otro lado, la hipermutación, debido a su naturaleza aleatoria, a menudo puede introducir cambios que deterioren a anticuerpos valiosos, degradando así la calidad total de la población de los anticuerpos. Por tanto la mutación puede favorecer como desfavorecer a la detección de patrones.

#### 4.10.3 Sistema Inmune Artificial e IDS.

En [Aic04] se diferencian dos categorías de sistemas inmunes artificiales basándose en el mecanismo que implementan. Por un lado los modelos basados en redes idiotípicas. En general, las poblaciones de anticuerpos están reguladas por otros anticuerpos que a su vez están conectados a otros anticuerpos y células del sistema inmune, formando lo que se llama una cadena o red idiotípica; una cadena sucesiva en que las poblaciones de unas células están reguladas por otras. Por otro lado, están los modelos de selección negativa, que consisten básicamente en distinguir lo propio (self) (usuarios legítimos, comportamiento normal, ficheros no corruptos, etc.) del resto (usuarios no autorizados, virus, anomalías, etc.).

El modelo de selección negativa, fue utilizado inicialmente como método de autenticación de ficheros para la detección de virus informáticos [For94]. En 1996, el grupo de Forrest realizó su primer experimento de detección de intrusos a partir de llamadas al sistema de procesos UNIX [For96]. El sistema recoge información de secuencias de comandos del agente de correo sendmail de UNIX, y la utiliza en el periodo de entrenamiento para definir lo que es propio.

Dasgupta y González también utilizaron selección negativa para la detección de intrusos [Das00][Das02a][Das02b]. Trabajaron con los datos DARPA de los laboratorios de Lincoln, y compararon el rendimiento de la selección positiva con la de la selección negativa para caracterizar lo propio. Como resultado, el método de

selección negativa era el más acertado. En [Gon02] combinan la selección negativa con técnicas de clasificación para la detección de anomalías. Para la selección negativa proponen un nuevo algoritmo llamado selección negativa de valor real (real-valued negative selection) para la representación del espacio propio/no propio. Al año siguiente, en [Gon03] utilizan muestras positivas (tráfico normal) para generar muestras negativas (anormal), las cuales se usan después como entrada a un algoritmo de clasificación. Comparan sus resultados con un sistema de detección de anomalías que utiliza mapas autoorganizativos.

#### 4.11 Máquinas de Soporte de Vectores “SVM”.

La teoría de las SVMs fue desarrollada inicialmente por V. Vapnik [Vap95] a principios de los años 80 y se centra en lo que se conoce como Teoría del Aprendizaje Estadístico. Realiza una clasificación lineal sobre vectores transformados a un espacio de dimensión superior, es decir, separa mediante un hiperplano en el espacio transformado.

El objetivo de las SVMs es encontrar el óptimo hiperplano que separe las dos clases y maximice el margen. Fig.12. Dado  $n$  muestras o vectores de entrenamiento representadas mediante pares  $(x_i, y_i)$ , donde  $y_i$  es la etiqueta de clase ( $y_i \in \{1, -1\}$ ) y  $x_i$  el vector de atributos ( $i = 1, \dots, n$ ); en el caso ideal (2 clases completamente separables) existe un número infinito de planos (o hiperplanos) que pueden separar las dos clases. El cálculo del hiperplano con margen óptimo es dado por la minimización de  $\|w\|^2$  obedeciendo a las siguientes restricciones:

$$y_i (x_i \cdot w + b) - 1 \geq 0, \quad \forall i$$

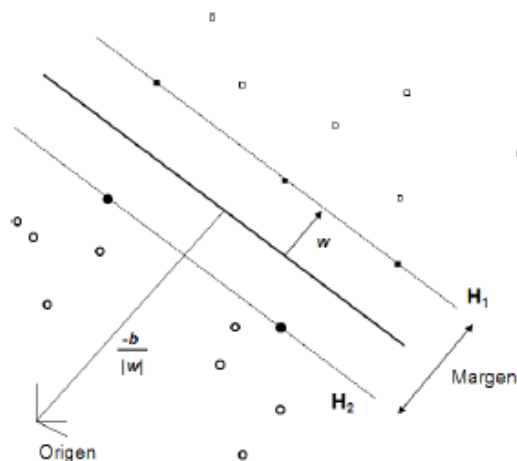


Figura 12 Clasificación de un conjunto de datos con SVM lineal

Donde  $w$  es un vector normal al hiperplano. Este un problema de optimización cuadrático y se puede solucionar Utilizando Multiplicadores de Lagrange, la solución es maximizar  $W(a)$  que viene dado por:

$$\sum_{i=1}^N a_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N a_i a_j y_i y_j (x_i \cdot x_j) \quad (4.10.1)$$

sujeto a:

$$\sum_i^N a_i y_i = 0$$

y

$$a_i \geq 0, \forall_i$$

Los elementos  $a_i$  son los multiplicadores de Lagrange, los vectores soporte corresponden a aquellos puntos donde  $a_i > 0$  mientras que  $a_i = 0$  indica los puntos de entrenamiento que están fuera del espacio limitado por  $H_1$  y  $H_2$ .

La mayoría de los datos no son solucionados con un hiperplano lineal, con lo cual se introducen variables de alargamiento del margen  $\xi_i$  que “relajen” las restricciones de la SVM lineal permitiendo algunos errores en el margen, así como penaliza los errores a través de la variable  $C$ .

$$x_i \cdot w + b \geq +1 - \xi_i \quad \text{para } y_i = +1$$

$$x_i \cdot w + b \leq -1 + \xi_i \quad \text{para } y_i = -1$$

Para obtener el hiperplano óptimo la solución es:

Minimizar:

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i$$

Ó maximizar la expresión (4.10.1) sujeto a:

$$\sum_i^N a_i y_i = 0,$$

y

$$0 \leq a_i \leq C, \forall_i$$

Otra solución de la no linealidad de los datos es la transformación de los datos a un espacio de dimensión muy alta a través de una función kernel. Se define por:

$$\mu \equiv \sum_{i=1}^N a_i y_i K(x_i, x) - b$$

Donde la minimización de los multiplicadores de Lagrange sigue siendo un problema cuadrático

$$\min \sum_{i=1}^N a_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N K(x_i, x_j) a_i a_j y_i y_j$$

Aplicando las condiciones de Karush-Kuhn-Tucker (KKT) en las ecuaciones anteriores se obtienen las siguientes condiciones para calcular el punto óptimo de un problema cuadrático positivo definido. De esta forma el problema cuadrático queda resuelto cuando para todo  $i$ :

$$\begin{aligned} a_i = 0 &\Leftrightarrow y_i u_i \geq 1, \\ 0 < a_i < C &\Leftrightarrow y_i u_i = 1, \\ a_i = C &\Leftrightarrow y_i u_i \leq 1 \end{aligned}$$

Pese a que en su forma más básica SVM induce separadores lineales, si el conjunto no es linealmente separable puede extenderse el algoritmo mediante una transformación no lineal  $\phi(x)$  a un nuevo espacio de características. La función permite transformar el espacio de características de entrada en un espacio de trabajo de mayor dimensionalidad donde intentar encontrar de nuevo el hiperplano óptimo. De esta forma se realiza una clasificación lineal en el nuevo espacio, que es equivalente a una clasificación no-lineal en el espacio original.

Las funciones núcleo (kernel functions, funciones kernel o simplemente kernels) son un tipo especial de función que permiten hacer la transformación del espacio de características de forma implícita durante el entrenamiento, sin necesidad de calcular explícitamente la función  $\phi$ , Schölkopf 2001 [Sch01], Shawe-Taylor [Shaw04]. Es lo que se conoce como kernel trick. Una vez que el hiperplano se ha creado, la función kernel se emplea para transformar los nuevos ejemplos al espacio de características para la clasificación. Formalmente, un kernel  $k$  es una función simétrica  $k(x_i, x_j) = \{\phi(x_i), \phi(x_j)\} = k(x_j, x_i)$  que puede ser interpretada como una medida de similitud entre dos vectores de características  $x_i$  y  $x_j$ . La selección del kernel apropiado es importante

ya que es éste el que define el espacio de trabajo transformado donde se llevará a cabo el entrenamiento y la clasificación.

*Lineal*

$$K(x_i, x_j) = \langle x_i, x_j \rangle$$

*Polinómico*

$$K(x_i, x_j) = (\gamma \langle x_i, x_j \rangle + r)^d, \gamma > 0$$

*Gaussiano o Radial Bassis Function (RBF)*

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0$$

*Sigmoide*

$$K(x_i, x_j) = \tanh(\gamma \langle x_i, x_j \rangle + r)$$

Donde  $\gamma$ ,  $d$  y  $r$  son los parámetros del kernel.

Cuando se utiliza una función kernel lineal, el espacio de los vectores y el espacio de características es el mismo [Wein04]. En un proceso de selección de funciones de Kernel, este tipo de función normalmente se utiliza como primera medida y posteriormente, se aplican métodos más complejos. Este método se ha empleado en diferentes herramientas bioinformáticas presentando resultados excelentes, especialmente cuando la dimensionalidad de los datos de entrada al modelo es grande y el número de ejemplos es pequeño [Ben08].

En una función kernel Polinómica se mapea los datos de entrada en un espacio de características con una dimensionalidad  $O(D^d)$ . Se debe tener precaución con este tipo de función porque por su flexibilidad en el manejo de las variables se puede facilitar negativamente el sobre-entrenamiento en conjuntos de gran dimensionalidad con un bajo número de ejemplos [Ben08].

En una función kernel Gaussiana se debe tener precaución porque por su flexibilidad en el manejo de las variables se puede facilitar negativamente el sobre entrenamiento en conjuntos de gran dimensionalidad con un bajo número de ejemplos [Ben08].



#### 4.11.1 Normalización.

Los clasificadores de margen amplio se caracterizan porque son sensibles a la manera como las características son escaladas, lo cual hace que sea esencial el proceso de normalización de los datos. La normalización puede ser aplicada en diferentes etapas del proceso, por ejemplo sobre las características de entrada o a la altura del Kernel (normalización en el espacio de características) o en ambas situaciones. Cuando las características son medidas en diferentes escalas y presentan diferentes rangos de valores posibles, es conveniente escalar los datos a un rango común. Los procesos de normalización hacen que los resultados difieran considerablemente cuando se utilizan las funciones lineales, polinomiales y gaussianas. En general, los procesos de normalización se asocian directamente con mejoras en el desempeño tanto en Kernels lineales como no lineales, acelerando los procesos de convergencia cuando se entrenan los clasificadores [Ben08].

#### 4.11.2 SMO Mínima Secuencia de Optimización.

El algoritmo Sequential Minimal Optimization SMO [Pla98], propone una manera rápida de entrenar las Máquinas de Soporte Vectorial mediante la solución de un algoritmo de programación dinámica en forma secuencial. Es un algoritmo que busca el Hiperplano de máximo margen que separe las instancias de clases de un dato [Pla98]. Es una técnica sencilla para la solución del problema cuadrático de las SVM, el algoritmo divide el problema en varios sub-problemas más simples. SMO trata de elegir siempre el menor problema cuadrático para ser optimizado en cada interacción y como este problema implica sólo dos multiplicadores de Lagrange, en cada iteración busca estos dos multiplicadores lleva a cabo la optimización y ajusta los valores de la SVM. La etapa de entrenamiento requiere un tipo de optimización conocido como Optimización Cuadrática Limitada que requiere tiempo y costo computacional.

#### 4.11.3 C-SVC.

LIBSVM es una biblioteca que implementa SVM desarrollada por Chin-Chung Chang [Cl01] para varios propósitos: estimación de la clasificación, la regresión y la distribución. El algoritmo de clasificación implementado en la biblioteca lleva el

nombre de C-SVC. Para la resolución de problemas de segundo grado C-SVC utiliza el mismo enfoque que SMO, es decir, descompone el conjunto de multiplicadores de Lagrange en un subconjunto más pequeño. Pero no sólo selecciona dos operadores arbitrariamente como en el SMO, sino que selecciona un subconjunto de tamaño variable [CI01]. Además de la selección de un subconjunto para su optimización, C-SVC también implementa las técnicas Shrinking y Caching para reducir el tiempo computacional. Shrinking trata de reducir el tamaño del problema a ser resuelto mediante la eliminación de multiplicadores de Lagrange de segundo grado que no se pueden cambiar en base a la heurística demostrada en [CI01]. La técnica de Caching simplemente almacena cálculos de matrices utilizados recientemente para su uso futuro, reduciendo parte de los cálculos del kernel realizados en las iteraciones futuras.

Originalmente SVM fue diseñado para problemas de clasificación binaria, para abordar el problema de clasificar en  $k$  clases, hay que transformar el problema de la clasificación multiclase en múltiples problemas de clasificación binaria [All01]. Hay dos aproximaciones básicas en este sentido: uno contra todos (one-against-all), donde se entrenan  $k$  clasificadores y cada uno separa una clase del resto, y la otra estrategia utilizada por SVC, “uno-contra-uno”, donde se han de entrenar  $(k(k-1))/2$  clasificadores y cada uno discrimina entre dos de las clases. Es importante notar que esta estrategia, al trabajar con menos muestras, tiene mayor libertad para encontrar una frontera que separe ambas clases. Respecto al coste de entrenamiento, es preferible el uso de uno contra todo puesto que sólo ha de entrenar  $k$  clasificadores.

#### 4.11.4 SVM e IDS

Eskin et al. utilizaron una SVM como complemento a sus métodos de clustering para el aprendizaje no supervisado [Esk02]. El trabajo descrito en [Amb03] está enfocado a la aplicación de clasificadores SVM múltiples, usando el método uno-contra-uno, para la detección de anomalías y también de uso indebido, identificando los ataques según su tipo forma precisa.

Mukkamala et al. utilizaron cinco SVM tradicionales anteriormente con el mismo fin; uno para identificar tráfico normal, y el resto para identificar cada uno de los cuatro tipos de ataques representados en el conjunto de datos de KDD Cup 99 [Muk02]. Los compararon con redes neuronales llegando a la conclusión de que los SVM

demostraban mejor desempeño. Este año han presentado un trabajo donde prueban que el uso conjunto de SVM y redes neuronales mejora la capacidad del IDS [Muk04].

## 4.12 Redes Neuronales

Son sistemas artificiales que van a copiar la estructura de las redes neuronales biológicas con el fin de alcanzar una funcionalidad similar.

Las redes neuronales artificiales [Hay94] tratan de emular tres conceptos claves:

- procesamiento paralelo, derivado de que los miles de millones de neuronas que intervienen, por ejemplo en el proceso de ver, están operando en paralelo sobre la totalidad de la imagen
- memoria distribuida, mientras que en un computador la información está en posiciones de memoria bien definidas, en las redes neuronales biológicas dicha información está distribuida por la sinapsis de la red, existiendo una redundancia en el almacenamiento, para evitar la pérdida de información en caso de que una sinapsis resulte dañada.
- adaptabilidad al entorno, por medio de la información de las sinapsis. Por medio de esta adaptabilidad se puede aprender de la experiencia y es posible generalizar conceptos a partir de casos particulares.

### 4.12.1 Arquitecturas de redes neuronales

Se denomina arquitectura a la topología, estructura o patrón de conexionado de una red neuronal. En una red neuronal artificial los nodos se conectan por medio de sinapsis, estando el comportamiento de la red determinado por la estructura de conexiones sinápticas. Estas conexiones sinápticas son direccionales, es decir, la información solamente puede propagarse en un único sentido (desde la neurona presináptica a la pos-sináptica). En general las neuronas se suelen agrupar en unidades estructurales que denominaremos *capas*. El conjunto de una o más capas constituye la red neuronal.

Se distinguen tres tipos de capas: de entrada, de salida y ocultas. Una *capa de entrada*, también denominada sensorial, está compuesta por neuronas que reciben datos o señales procedentes del entorno. Una *capa de salida* se compone de neuronas que proporcionan la respuesta de la red neuronal. Una *capa oculta* no tiene una conexión directa con el entorno, es decir, no se conecta directamente ni a órganos sensores ni a efectores. Este

tipo de capa oculta proporciona grados de libertad a la red neuronal gracias a los cuales es capaz de representar más fehacientemente determinadas características del entorno que trata de modelar.

Así considerando su estructura podemos hablar de *redes monocapa*, compuestas por una única capa de neuronas o *redes multicapa*, las neuronas se organizan en varias capas. Teniendo en cuenta el flujo de datos, podemos distinguir entre *redes unidireccionales (feedforward)* y *redes recurrentes o realimentadas (feedback)*. Mientras que en las redes unidireccionales la información circula en un único sentido, en las redes recurrentes o realimentadas la información puede circular entre las distintas capas de neuronas en cualquier sentido, incluso en el de salida-entrada.

El perceptrón es quizás la forma más simple de una red neuronal que se puede utilizar para la clasificación de clases o conceptos que sean linealmente separables, es decir que las muestras positivas y negativas de la clase se pueden separar mediante un hiperplano en el espacio de características  $X$ , en las Fig 13 y 14 se muestra un ejemplo para dimensión 2.

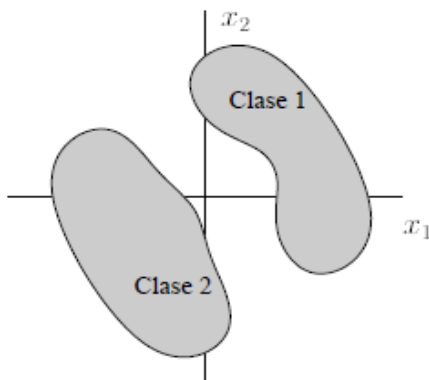


Figura 13 Clase linealmente Separable.

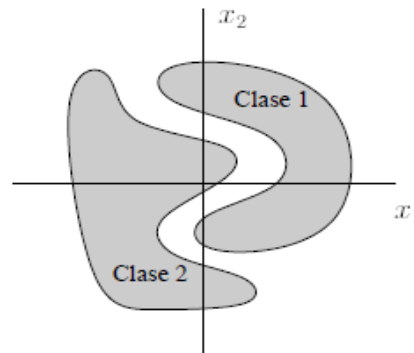


Figura 14 Clase no linealmente Separable.

#### 4.12.2 Perceptrón Multicapa

Un perceptrón multicapa está compuesto por una capa de entrada, una capa de salida y una o más capas ocultas; aunque se ha demostrado que para la mayoría de problemas bastará con una sola capa oculta [Fun89] [Hor89]. En la figura 15 podemos observar un perceptrón típico formado por una capa de entrada, una capa oculta y una de salida.

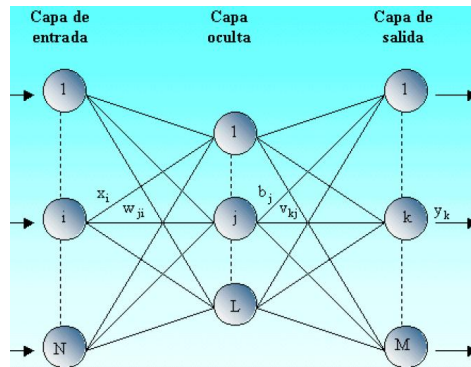


Figura 15 Perceptrón con una capa oculta

En este tipo de arquitectura, las conexiones entre neuronas son siempre hacia delante, es decir, las conexiones van desde las neuronas de una determinada capa hacia las neuronas de la siguiente capa; no hay conexiones laterales --esto es, conexiones entre neuronas pertenecientes a una misma capa (ni conexiones hacia atrás), esto es, conexiones que van desde una capa hacia la capa anterior. Por tanto, la información siempre se transmite desde la capa de entrada hacia la capa de salida. En el presente documento, hemos considerado  $w_{ji}$  como el peso de conexión entre la neurona de entrada  $i$  y la neurona oculta  $j$ , y  $v_{kj}$  como el peso de conexión entre la neurona oculta  $j$  y la neurona de salida  $k$ .

#### *Algoritmo backpropagation*

En el algoritmo backpropagation podemos considerar, por un lado, una etapa de funcionamiento donde se presenta, ante la red entrenada, un patrón de entrada y éste se transmite a través de las sucesivas capas de neuronas hasta obtener una salida y, por otro lado, una etapa de entrenamiento o aprendizaje donde se modifican los pesos de la red de manera que coincida la salida deseada por el usuario con la salida obtenida por la red ante la presentación de un determinado patrón de entrada.

#### *Etapa de funcionamiento*

Cuando se presenta un patrón  $p$  de entrada  $X_p: x_{p1}, \dots, x_{pi}, \dots, x_{pN}$ , éste se transmite a través de los pesos  $w_{ji}$  desde la capa de entrada hacia la capa oculta. Las neuronas de esta capa intermedia transforman las señales recibidas mediante la aplicación de una función de activación proporcionando, de este modo, un valor de salida. Este se transmite a través de los pesos  $v_{kj}$  hacia la capa de salida, donde aplicando la misma operación que en el caso anterior, las neuronas de esta última capa proporcionan la

salida de la red. Este proceso se puede explicar matemáticamente de la siguiente manera:

La entrada total o neta que recibe una neurona oculta  $j$ ,  $net_{pj}$ , es:

$$net_{pj} = \sum_{i=1}^N w_{ji} x_{pi} + \theta_j$$

donde  $\theta$  es el umbral de la neurona que se considera como un peso asociado a una neurona ficticia con valor de salida igual a 1.

El valor de salida de la neurona oculta  $j$ ,  $b_{pj}$ , se obtiene aplicando una función  $f(\cdot)$  sobre su entrada neta:

$$b_{pj} = f(net_{pj})$$

De igual forma, la entrada neta que recibe una neurona de salida  $k$ ,  $net_{pk}$ , es:

$$net_{pk} = \sum_{j=1}^L v_{kj} b_{pj} + \theta_k$$

Por último, el valor de salida de la neurona de salida  $k$ ,  $y_{pk}$ , es:

$$y_{pk} = f(net_{pk})$$

### *Etapa Aprendizaje*

En la etapa de aprendizaje, el objetivo que se persigue es hacer mínima la discrepancia o error entre la salida obtenida por la red y la salida deseada por el usuario ante la presentación de un conjunto de patrones denominado grupo de entrenamiento. Por este motivo, se dice que el aprendizaje en las redes backpropagation es de tipo supervisado, debido a el usuario (o supervisor) determina la salida deseada ante la presentación de un determinado patrón de entrada.

La función de error que se pretende minimizar para cada patrón  $p$  viene dada por:

$$E_p = \frac{1}{2} \sum_{k=1}^M (d_{pk} - y_{pk})^2$$

donde  $d_{pk}$  es la salida deseada para la neurona de salida  $k$  ante la presentación del patrón  $p$ . A partir de esta expresión se puede obtener una medida general de error mediante:

$$E = \sum_{p=1}^P E_p$$

La base matemática del algoritmo backpropagation para la modificación de los pesos es la técnica conocida como gradiente decreciente [Rum86]. Teniendo en cuenta que  $E_p$  es función de todos los pesos de la red, el gradiente de  $E_p$  es un vector igual a la derivada parcial de  $E_p$  respecto a cada uno de los pesos. El gradiente toma la dirección que determina el incremento más rápido en el error, mientras que la dirección opuesta -- es decir, la dirección negativa--, determina el decremento más rápido en el error. Por tanto, el error puede reducirse ajustando cada peso en la dirección:

$$-\sum_{P=1}^P \frac{\partial E_p}{\partial w_{ji}}$$

#### 4.12.3 Redes de función de Base Radial

Una RBF (Radial Basis Function) es muy similar en la arquitectura a un Perceptrón Multicapa, pero su forma de aprendizaje es diferente. Este tipo de redes se caracteriza por tener un aprendizaje o entrenamiento híbrido. La arquitectura de estas redes se caracteriza por la presencia de tres capas: una de entrada, una única capa oculta y una capa de salida.

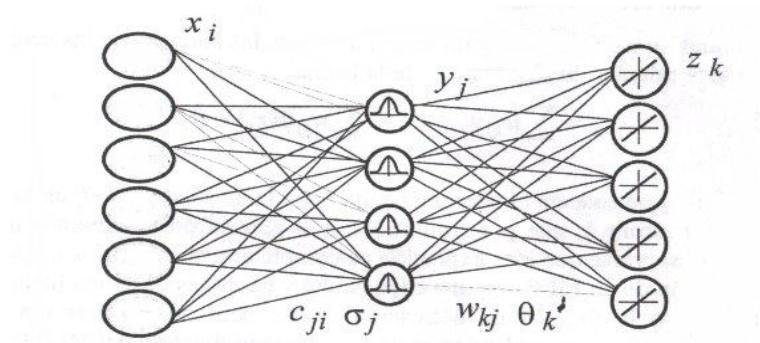


Figura 16 Arquitectura típica de una RBF

Aunque la arquitectura pueda recordar a la de un perceptrón multicapa, la diferencia fundamental está en que las neuronas de la capa oculta en vez de calcular una suma ponderada de las entradas y aplicar una función de activación sigmoide, estas neuronas calculan la distancia euclídea entre el vector de pesos sinápticos (que recibe el nombre en este tipo de redes de centro o centroide) y la entrada y sobre esa distancia se aplica una función de tipo radial con forma gaussiana.

*Aprendizaje*

Para el aprendizaje de la capa oculta, hay varios métodos, siendo uno de los más conocidos el algoritmo denominado k-medias (k-means) que es un algoritmo no supervisado de clustering. k es el número de grupos que se desea encontrar, y se corresponde con el número de neuronas de la capa oculta, que es un parámetro que hay que decidir de antemano. El algoritmo se plantea como sigue:

1. Inicializar los pesos (los centros) en el instante inicial. Una inicialización típica es la denominada k-primera mediante la cual los k centros se hacen iguales a las k primeras muestras del conjunto de datos de entrenamiento  $\{\mathbf{x}_p\}_{p=1..N}$

$$\mathbf{c}_1 = \mathbf{x}_1, \mathbf{c}_2 = \mathbf{x}_2, \dots, \mathbf{c}_k = \mathbf{x}_k,$$

2. En cada iteración, se calculan los dominios, es decir, se reparten las muestras entre los k centros. Esto se hace de la siguiente manera: Dada una muestra  $\mathbf{x}_j$  se calcula las distancias a cada uno de los centros  $\mathbf{c}_k$ . La muestra pertenecerá al dominio del centro cuya distancia calculada sea la menor
3. Se calculan los nuevos centros como los promedios de los patrones de aprendizaje pertenecientes a sus dominios. Viene a ser como calcular el centro de masas de la distribución de patrones, tomando que todos pesan igual.
4. Si los valores de los centros varían respecto a la iteración anterior se vuelve al paso 2, si no, es que se alcanzó la convergencia y se finaliza el aprendizaje

Una vez fijados los valores de los centros, sólo resta ajustar las anchuras de cada neurona. Las anchuras son los parámetros sigma que aparecen en cada una de las funciones gaussianas y reciben ese nombre por su interpretación geométrica, dan una medida de cuando una muestra activa una neurona oculta para que de una salida significativa, normalmente se toma el criterio de que para cada neurona se toma como valor sigma la distancia al centro más cercano.

Finalmente, se entrena la capa de salida. El entrenamiento de esta capa se suele usar un algoritmo parecido al que se usa para la capa de salida del MLP. La actualización de los pesos viene dada por la expresión:

$$z_k = \sum_j w_{kj} \phi(r_j) + \theta_k$$

$$\delta w_{kj}(t) = \theta(d_k - z_k) \phi(r_j)$$



#### 4.12.4 Redes Neuronales e IDS

Se han realizado numerosos trabajos con redes neuronales artificiales en detección de intrusos tratando de dar una alternativa a los sistemas expertos gracias a su flexibilidad y adaptación a los cambios naturales que se pueden dar en el entorno y, sobre todo, a la capacidad de detectar instancias de los ataques conocidos. La mayor deficiencia que tienen las redes neuronales es que son un modelo no descriptivo, es decir; actúan como una caja negra sin que se pueda conocer la razón de la decisión tomada.

El primer modelo de detección de intrusos basado en redes neuronales lo realizaron Fox et al. como método para crear perfiles de comportamiento de usuarios [Fox90]. Al igual que en [Deb92a], utilizan redes neuronales para predecir el siguiente comando basado en una secuencia de comandos previos ejecutados por un usuario. El aprendizaje lo realizan mediante redes neuronales recurrentes (parte de la salida se realimenta como entrada a la red en la siguiente iteración) por lo que la red está continuamente observando y tiene la capacidad de “olvidar” comportamientos antiguos. Debar y Dorizzi presentan un sistema de filtrado basado en redes neuronales recurrentes que actúa para filtrar los datos que no se corresponden con la tendencia observada en el comportamiento de las actividades de usuarios [Deb92b].

Ryan et al. desarrollaron NNID (Neural Network Intrusion Detection) para la identificación de usuarios legítimos basado en la distribución de los comandos que ejecutaban. Escogieron una arquitectura de red neuronal multicapa de tipo back-propagation de tres capas para su cometido [Rya98]. David Endler utilizó un perceptrón multicapa tanto para la detección de uso indebido como para la detección de anomalías a partir de datos de auditoría procedentes del BSM (Basic Security Module) de Solaris [End98].

Lippmann y Cunningham realizaron un proyecto que mejoraba el rendimiento de la detección de ataques de tipo U2R realizados mediante el uso de palabras clave [Lip99]. Una vez obtenidas las palabras clave, se usaba una red de tipo perceptrón multicapa (sin ninguna capa oculta) para la detección de ataques. Más tarde, se utilizó otra red neuronal similar para su clasificación. Una red de tipo perceptrón multicapa (sin ninguna capa oculta) mide inicialmente el número de palabras clave, proporcionando una estimación de la probabilidad posterior de un ataque en cada sesión telnet. La otra

red, del mismo tipo, se utilizaba posteriormente para tratar de clasificar ataques conocidos y de esa manera facilitar el nombre de dicho ataque. Este mismo año, Ghosh y Schwartzbard presentan un trabajo muy similar a los anteriores, pero en lugar de utilizar redes neuronales para crear perfiles del comportamiento de usuarios, utilizan la red para crear perfiles del comportamiento del software de modo que tratan de distinguir entre comportamiento de software normal y malicioso [Gho99]. Utilizan una red neuronal de tipo backpropagation (perceptrón feed-forward multicapa) con el fin de generalizar datos incompletos y posteriormente realizar la clasificación.

En la Universidad de Ohio se ha desarrollado un IDS de red llamado INBOUNDS (Integrated Network-Based Ohio University Network Detective Service) donde un módulo de detección de anomalías basado en análisis estadístico se ha sustituido por otro que utiliza mapas autoorganizativos [Ram03].

#### 4.13 Modelos Ocultos de Markov

Los HMM fueron inicialmente estudiados e introducidos por L. E. Baum en los años 70's [Bau72], Baum propone este modelo como un método estadístico de estimación de las funciones probabilísticas de una cadena de Markov.

Un modelo oculto de markov es un autómata de estados finitos que produce como salida una secuencia de símbolos observables. Se llaman ocultos porque existe un proceso de probabilidad subyacente que no es observable, pero afecta a la secuencia de eventos observados.

Con el objetivo de comprender el funcionamiento de los modelos HMM, a continuación se presentan en detalle tanto el conjunto de “elementos” que lo componen como el procedimiento general para la generación de observaciones. Para ello se utilizará un ejemplo sencillo conocido como el modelo de Urnas y Bolas presentado en [Rab90].



Figura 17 Modelo de Urna

Supóngase que se dispone por un lado de un conjunto de  $N$  urnas y por otro de un conjunto de  $M$  bolas de distintos colores, de forma que cada urna contiene un

número de bolas, el cual puede ser distinto de una urna a otra. Por tanto en cada urna se tendrá una distribución de probabilidad distinta asociada al color de las bolas. Para calcular dicha distribución bastaría con utilizar la definición de probabilidad que para cada color es el número de bolas de ese color entre el número total de bolas en esa urna. Supongamos que existe un proceso que se repite un número finito de veces por el cual, inicialmente de forma aleatoria una urna es escogida y de ella se extrae una bola la cual anotamos su color y dicha bola es devuelta a la urna, así  $T$  veces. El único dato accesible desde el exterior es el color de la bola elegida, con lo cual el fenómeno observable será una secuencia de colores. Si esta secuencia de colores se le presenta a una persona ajena al proceso de extracción de las bolas, ésta únicamente verá la secuencia de colores ignorando la secuencia de urnas involucradas en el proceso.

De esta manera se puede decir que las urnas corresponderían a la secuencia de estados del modelo y permanecen ocultos al observador. De esta sencilla manera se modela un Modelo oculto de Markov para explicar el proceso de tal forma que contenga  $N$  estados, cada uno de los cuales se corresponde a una urna. Cada estado tiene asociada unas probabilidades de selección de cada uno de los colores y cada pareja de estados una probabilidad de transición del primero al segundo y viceversa. Utilizando el modelo se puede evaluar, no sólo la secuencia de colores obtenida, sino que además se obtendrá una estimación de la secuencia de urnas que han intervenido en el proceso.

#### 4.13.1 Arquitectura HMM.

Un HMM puede ser representado como un grafo dirigido de transiciones/emisiones como se ilustra en la figura 18. La arquitectura específica que permita modelar de la mejor forma posible las propiedades observadas depende en gran medida de las características del problema. Las arquitecturas más usadas son:

- *Ergódicas o completamente conectadas* en las cuales cada estado del modelo puede ser alcanzado desde cualquier otro estado en un número finito de pasos.
- *Izquierda-derecha, hacia adelante o Bakis* las cuales tienen la propiedad de que en la medida que el tiempo crece se avanza en la secuencia de observación asociada  $O$ , y en esa misma medida el índice que señala el estado del modelo permanece o crece, es decir, los estados del sistema van

de izquierda a derecha. En secuencias biológicas y en reconocimiento de la voz estas arquitecturas modelan bien los aspectos lineales de las secuencias.

- *Izquierda-derecha paralelas*, son dos arquitecturas *izquierda-derecha* conectadas entre sí.

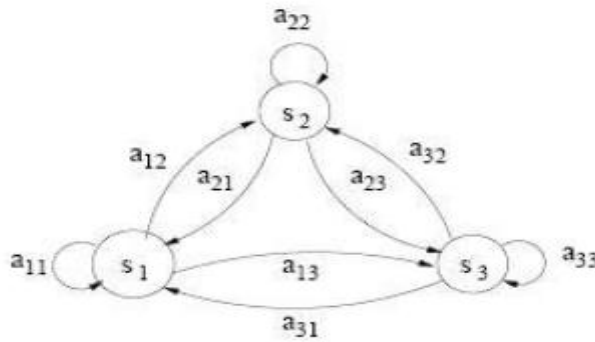


Figura 18 Estructura HMM Ergódico

#### 4.13.2 Modelos Ocultos de Markov discretos

Un modelo oculto de markov discreto queda definido en función de los siguientes elementos [Rab90]

1. Número de estados del modelo. “N”
2. Número de observaciones distintas “M”
3. Conjunto finito de estados  $S = \{Q_1, Q_2, \dots, Q_n\}$
4. Conjunto discreto de observaciones  $O = \{O_1, O_2, \dots, O_m\}$
5. Matriz de probabilidades de transición, esta matriz es cuadrada de tamaño N y sus elementos se corresponden a la probabilidad de transición de un estado a otro.  $A = \{a_{ij}\}$ .
6.  $B = \{b_i(k)\}$  Matriz de probabilidades de observación. Su tamaño es de  $N \times M$ . Probabilidad de que se produzca el símbolo asociado a al índice k cuando se está en el estado  $Q_i$ .
7.  $\Pi = \{\pi\}$  Probabilidad estado inicial, siendo  $\pi_i$  la probabilidad de que el estado inicial sea  $Q_i$ .

Un modelo HMM discreto queda identificado por tanto por su conjunto de parámetros, probabilidades de transición, probabilidades de observación y probabilidad del estado inicial, el cual se denota habitualmente como  $\lambda = (A, B, \Pi)$ .

Existen 3 problemas relacionados con los HMM [Rab90]:

Problema 1. Dada una secuencia de observaciones  $O_1, O_2, \dots, O_m$  y un modelo  $\lambda = (A, B, \Pi)$  ¿cómo se puede calcular eficientemente la probabilidad de la secuencia de observaciones de haber sido generada por el modelo  $\lambda$ , es decir  $P(O|\lambda)$ ?

Problema 2. Dada la secuencia de observaciones  $O_1, O_2, \dots, O_m$  y el modelo  $\lambda$ , ¿cómo se selecciona una secuencia de estados  $Q = q_1, q_2, \dots, q_T$  que sea óptima y que explica de la mejor manera posible la secuencia de observaciones?

Problema 3. ¿Cómo se pueden ajustar los parámetros del modelo  $\lambda = (A, B, \pi)$  para maximizar  $P(O|\lambda)$ ? También conocido como el problema del aprendizaje. A continuación se muestran las soluciones a estos tres problemas, detallados en [Rab90].

**Solución al Problema 1:** Algoritmo *forward-backward* [Bau67] que permite resolver el problema de forma eficiente. La idea propuesta en este algoritmo es considerar la secuencia de observaciones hasta un instante de tiempo  $t$  y calcular la siguiente probabilidad:

$$\alpha_{i,t} = P(O_1, O_2, \dots, O_t, Q_t = q_i | \lambda)$$

es decir la probabilidad conjunta de la secuencia de observaciones hasta el periodo  $t$  y el estado  $Q_i$  en el instante  $t$  dado el conjunto de parámetros del modelo. Esta variable auxiliar puede calcularse fácilmente de forma inductiva teniendo en cuenta el vector de probabilidades iniciales  $\Pi = (\pi_1, \dots, \pi_n)$  y la matriz de probabilidades de transición  $A$ .

El algoritmo se resume en los siguientes pasos:

1.  $\alpha_1(i) = \pi_i b_i(O_1)$  para  $1 \leq i \leq N$

2. para  $t = 1, 2, \dots, T-1$  para  $1 \leq j \leq N$

$$\alpha_t(j) = \left[ \sum_{i=1}^N \alpha_{t-1}(i) a_{ij} \right] b_j(O_{t+1})$$

3. Por tanto  $P(O|\lambda) = \sum_{i=1}^N \alpha_{T-1}(i)$

donde  $b_i(O_t)$  representa la probabilidad de emisión de una determinada observación en el instante de tiempo  $t$  en el estado  $Q_i$ .

Los tres pasos presentados se corresponden con la versión *forward* del algoritmo *forward-backward*.

De forma análoga es posible definir la variable *backward*  $\beta_i$  que representa la probabilidad de la observación parcial de la secuencia desde  $t+1$ , hasta el final dado el estado  $q_i$  en el momento  $t$ , y el modelo  $\lambda$ . Es decir,

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \dots, O_T \mid i_t = q_i, \lambda)$$

Esta variable también puede ser calculada de forma inductiva comenzando por  $\beta_{i,T}$  y retrocediendo hasta  $\beta_{i,1}$  teniendo en cuenta la matriz de transición.

**Solución al Problema 2:** El problema 2 pretende descubrir la sucesión de estados ocultos que mejor describe una secuencia de observaciones.

Esto se soluciona gracias al algoritmo de Viterbi [For73] Para encontrar la mejor secuencia de estados  $q_1, \dots, q_t$  dada una secuencia de observaciones  $O_1, O_2, \dots, O_m$  se define la cantidad:

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_t} P(q_1, q_2, \dots, q_t = i, O_1, O_2, \dots, O_t \mid \lambda) \quad (2.1)$$

Esta cantidad representa la más alta probabilidad a través de un solo camino, en el instante  $t$ , que considera las primeras  $t$  observaciones y termina en el estado  $S_i$ .

Por inducción se tiene que:

$$\delta_{t+1}(j) = [\max_j \gamma_t(i) a_{ij}] b_j(O_{t+1}) \quad (2.2)$$

Para poder tener la secuencia de estados se debe llevar la cuenta de los argumentos que maximizan a 2.2 para cada  $t$  y  $j$ . Esto se hace a través del vector  $\psi_t(j)$ , siguiendo el procedimiento recursivo descrito a continuación:

1. Inicialización:

$$\begin{aligned} \delta_1(i) &= \pi_i b_i(O_1) && \text{para } 1 \leq i \leq N \\ \psi_1(i) &= 0 \end{aligned}$$

2. Recursión:

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t) \quad \text{para } 2 \leq t \leq T$$

$$\text{para } 1 \leq j \leq N$$

$$\psi_t(j) = \operatorname{argmax}_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \quad \text{para } 2 \leq t \leq T$$

$$\text{para } 1 \leq j \leq N$$

3. Terminación

$$p^* = \max_{1 \leq i \leq N} [\delta_T(i)]$$

$$q^* = \operatorname{argmax}_{1 \leq i \leq N} [\psi_T(i)]$$

4. Secuencia de Estados.

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \quad t = T-1, T-2, \dots, 1$$

**Solución al Problema 3 :** Consiste en el cálculo de los parámetros que caracterizan el modelo. Dados un conjunto de datos y una colección de secuencias observables, se determina el HMM que con mayor probabilidad ha generado la secuencia. Este problema se resuelve comúnmente con el algoritmo *Baum-Welch* [Bau67][Bau72].

Aquí el problema que tenemos es que queremos estimar los parámetros del modelo  $\lambda$  (A, B,  $\Pi$ ) de forma que maximicemos  $P(O/\lambda)$ . Sin embargo, no existe ningún método conocido que permita obtener analíticamente el juego de parámetros que maximice la secuencia de observaciones. Por otro lado, podemos determinar este juego de características de modo que su verosimilitud encuentre un máximo local mediante la utilización de procedimientos iterativos como el del método de Baum-Welch, este no es más que un algoritmo E-M [Dem77][Pre04] aplicado a los HMM; o bien mediante la utilización de técnicas de gradiente.

Al proceso de ajuste de los parámetros se le conoce como entrenamiento o aprendizaje del modelo oculto de Markov.

Para describir el procedimiento se define un nuevo parámetro,  $\xi_t(i,j)$ , como la probabilidad de encontrarnos en el estado  $i$  en el instante  $t$ , y en el estado  $j$  en el instante  $t+1$ , para un modelo y una secuencia de observación dados:

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda)$$

Utilizando las probabilidades de los métodos forward y backward podemos escribir  $\xi_t(i,j)$  con la siguiente fórmula

$$\xi_t = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)} = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}$$

Suponiendo  $\gamma_t(i)$  la probabilidad de encontrarnos en el estado  $i$  en el instante  $t$ , para la secuencia de observaciones completa y el modelo dados; por lo tanto, a partir de  $\xi_t(i,j)$  podemos calcular  $\gamma_t(i)$  con solo realizar el sumatorio para toda  $j$ , de la forma:

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i,j)$$

Realizando el sumatorio de  $\gamma_t(i)$  para todo  $t$ , obtenemos un resultado que puede ser interpretado como el numero esperado de veces (en el tiempo) que estamos en el estado  $i$  o de manera equivalente, numero esperado de transiciones realizadas desde el estado  $i$  (excluyendo el instante  $t=T$  del sumatorio). De forma análoga, el sumatorio de  $\xi_t(i,j)$  en  $t$  (desde  $t=1$  hasta  $t=T-1$ ) puede ser interpretado como el numero esperado de transiciones desde el estado  $i$  al estado  $j$ .

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{número esperado de transiciones desde } S_i$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{número esperado de transiciones de } S_i \text{ a } S_j$$

Con lo anterior, Baum propone las siguientes fórmulas de re-estimación quedando:

$\bar{\pi}_i$  = frecuencia esperada del estado  $S_i$  en  $t = 1$  (número de veces visitado)

$$\bar{\pi}_i = \gamma_t(i) \quad (1)$$

$$\bar{a}_{ij} = \frac{\text{N}^\circ \text{ esperado de transiciones de } S_i \text{ a } S_j}{\text{N}^\circ \text{ esperado de transiciones de } S_i}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \sum_{k=1}^N \xi_t(i,k)} \quad (2)$$

$$\bar{b}_j(i) = \frac{\text{N}^\circ \text{ esperado de } S_i \text{ y observar } v_k}{\text{N}^\circ \text{ esperado de visitas a } S_j}$$

$$\bar{b}_j(i) = \frac{\sum_{t=1}^{T-1} \sum_{k=1}^N \xi_t(i,k) Y_t(i)}{\sum_{t=1}^{T-1} \sum_{k=1}^N \xi_t(i,k)} \quad (3)$$

Si usamos los parámetros del modelo inicial  $\lambda = (A,B, \pi)$  para calcular los valores de las formulas anteriores (1,2,3) para obtener  $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi}, )$ , de acuerdo con Baum se puede probar que:



El modelo inicial  $\lambda$  es un punto crítico de la función de probabilidad  $P(O|\lambda)$ , en cuyo caso  $\lambda = \bar{\lambda}$  o bien

El modelo  $\bar{\lambda}$  es más probable que el modelo  $\lambda$  en el sentido de  $P(O|\bar{\lambda}) > P(O|\lambda)$ . Basándose en el procedimiento anterior, si se reemplaza iterativamente  $\bar{\lambda}$  por  $\lambda$  y repetimos la re-estimación, se puede mejorar la probabilidad de que  $O$  sea observado por el modelo hasta alcanzar un límite. Al resultado final de la re-estimación se le llama “estimado de mayor posibilidad” del modelo oculto de Markov.

#### 4.13.3 Modelos de Markov e IDS.

Nassehi propuso en 1998 el uso de cadenas de Markov para la detección de anomalías. Construyó su cadena de Markov con un tamaño de ventana unitario [Nas98]. En 1999, Lane investigó el uso de modelos ocultos de Markov, conocidos como HMM (Hidden Markov Models), para crear perfiles de usuarios y medidas de similitud [Lane99]. En esa misma fecha, Warrender, del grupo de Stephanie Forrest, utilizó HMM como el modelo subyacente para la detección utilizando llamadas del sistema [War99]. Sus resultados resultaron ser muy parecidos a los experimentos que realizaron más adelante Jha et al., con un método para la detección de anomalías pero basada en cadenas de Markov, en las que dichas cadenas se utilizan para construir el clasificador [Jha01]. También se puede encontrar un trabajo muy parecido al de Warrender en [Gao02].

#### 4.14 Discretización

El proceso de discretización entra en la parte de preprocesado de los datos en el proceso de KDD. Antes de tratar el concepto de discretización vamos a exponer los tipos de datos que existen. Generalmente se hace la distinción en:

*Cuantitativas.* Se distinguen a su vez en

- **Discretas** Un atributo discreto tiene un número finito o contable de valores. En general se representa como números enteros. Atributos binarios son un caso especial de ellos.

- Continuas Un atributo continuo tiene un número infinito de valores posibles. Es representado por números reales o de punto flotante. Se pueden obtener tan precisos como sea el instrumento de medición.

*Cualitativas o Categóricas.* Se pueden distinguir

- Nominales. No tienen orden significativo, nombran el objeto al que se refieren
- Ordinales. Tienen orden definido se puede establecer un orden en sus valores

La discretización, es de especial importancia en Inteligencia Artificial, pues permite que muchos algoritmos de aprendizaje ideados para funcionar con atributos nominales o categóricos puedan también utilizarse con conjuntos de datos que incluyen valores numéricos, algo esencial en la resolución de problemas reales. Un gran número de algoritmos de aprendizaje operan exclusivamente con espacios discretos, sin embargo, muchas bases de datos contienen atributos de dominio continuo, lo que hace imprescindible la aplicación previa de algún método que reduzca la cardinalidad del conjunto de valores que estas características pueden tomar, dividiendo su rango en un conjunto finito de intervalos. Esta transformación de atributos continuos en discretos se denomina *discretización*.

La discretización de los valores no sólo permite construir modelos de clasificación más compactos y sencillos, que resultan más fáciles de comprender, comparar, utilizar y explicar, sino que permite mejorar la precisión del clasificador y hace más rápido el aprendizaje.

Existen diversas clasificaciones para los métodos de discretización tales como, supervisados y no supervisados, locales también llamados dinámicos y globales o estáticos.

Los métodos supervisados solo son aplicables cuando se trabaja con datos que están divididos en clases. Estos métodos utilizan la información de la clase cuando se selecciona los puntos de corte en la discretización. Pueden además ser caracterizados como *basados en error*, *basados en entropía* o *basados en estadísticas*. Los métodos basados en error aplican un clasificador a los datos transformados y seleccionan los intervalos que minimizan el error en el conjunto de entrenamiento. En contraste, los métodos basados en entropía y los basados en estadísticas evalúan respectivamente la entropía de la clase o alguna otra estadística con respecto a la relación entre los

intervalos y la clase. Por otra parte los métodos no supervisados no utilizan la información de la clase.

Los métodos globales usan todo el espacio de instancias para el proceso de discretización. En cambio los métodos locales usan solo un subconjunto de las instancias para el proceso de discretización. Se relacionan con la discretización dinámica. Un atributo cualquiera puede ser discretizado en distintos intervalos (árboles). Las técnicas globales son más eficientes, porque solamente se usa una discretización a través de todo el proceso de data mining, pero las técnicas locales podrían provocar el descubrimiento de puntos de corte más útiles.

Para llevar a cabo nuestros experimentos se eligieron dos técnicas diferentes de discretización. Una de ellas llamada El método de Intervalos de igual Frecuencia -en [Dou95] se repasan distintos métodos de discretización utilizados en inteligencia artificial y aprendizaje automático- que es uno de los más sencillos algoritmos de discretización que existen. Básicamente este algoritmo opera de la siguiente manera, requiere que los valores de los atributos sean ordenados, suponiendo que el atributo a discretizar tiene  $m$  valores distintos, este discretizador divide el dominio de cada variable en  $n$  partes, donde cada parte tiene

$$\frac{m}{n}$$

valores continuos del atributo. Se trata de un algoritmo no supervisado como ya se ha mencionado anteriormente esta técnica no tiene en cuenta la clase.

Por contraposición el otro tipo de algoritmo escogido ha sido un algoritmo supervisado concretamente el de Fayyad & Irani [Fay93]. El método de discretización Fayyad e Irani's trabaja sin un número predefinido de intervalos. Para ello se dividen recursivamente las características dentro de intervalos en cada fase minimizando la entropía respecto a los intervalos y la información requerida para especificar esos intervalos. Se detiene la separación cuando la entropía no puede reducirse más.

Los métodos basados en entropía utilizan la información existente de la clase en los datos. La entropía (o contenido de información) es calculada en base a la clase. Intuitivamente, encuentra la mejor partición de tal forma que las divisiones sean lo más

puras posible, i.e. la mayoría de los valores en una división corresponden a la misma clase. Formalmente, es caracterizado por encontrar la partición con la máxima ganancia de información. Es un método de discretización supervisado, global y estático.

Fayyad usa la heurística de mínima entropía para discretizar el rango de los atributos de valores continuos en múltiples intervalos. Esta técnica utiliza el criterio de longitud mínima de descripción para controlar el número de intervalo [Fay93].

Para explicar este método se tomará como base la heurística de minimización de información de entropía para discretización binaria (división de dos intervalos), ésta será extendida a múltiples intervalos en vez de sólo dos.

La discretización binaria se basa en un valor de umbral  $T$ , que determina la separación para el atributo cuyo valor  $A$  es continuo. El valor se asigna a la rama izquierda si  $A \leq T$ , por el contrario, si  $A > T$ , se asigna a la rama derecha. El valor de umbral  $T$  se considera un punto de corte. A partir de un conjunto  $S$  con  $N$  muestras, para cada atributo de valor continuo  $A$  se toma el mejor punto de corte  $T_A$ , evaluando todos los posibles candidatos como puntos de corte. Los candidatos se obtienen mediante los puntos medios entre cada par sucesivo de muestras, la secuencia de muestras debe estar ordenada ascendentemente. Así para cada atributo de valor continuo, se tendrán  $N - 1$  evaluaciones. Para cada candidato de punto de corte  $T$ , los datos son divididos en dos conjuntos, y se calcula la entropía de clase de la partición resultante.

La fórmula usada para calcular la entropía es:

$$Ent(S) = \sum_{i=1}^k P(C_i, S) \log(P(C_i, S))$$

donde  $P(C_i, S)$  es el número de casos correspondientes a la clase  $C_i$  sobre el total de casos en  $S$ . Para evaluar cada punto de corte se calcula la entropía sobre ambas particiones ( $S_1$  y  $S_2$ ) de manera ponderada:

$$E(A, T; S) = \frac{S_1}{S} Ent(S_1) + \frac{S_2}{S} Ent(S_2)$$

La extensión de la discretización con múltiples intervalos es simple, la idea se basa en hacer recursivo el proceso de cortes binarios, aplicando un criterio para decidir cuando abstenerse de seguir aplicando más particiones. El criterio de paro, se basa en el principio de MDL (Longitud de Descripción Mínima) y determina si se debe aceptar o

no el corte propuesto. A partir del punto de corte T para el conjunto S compuesto de N ejemplos será aceptado mediante el criterio MDLP si y sólo si:

$$[Ent(S) - E(A, T; S)] > \left[ \frac{\log_2(N-1)}{N} + \log_2(3^k - 2) - kEnt(S) + k_1Ent(S_1) + k_2Ent(S_2) \right]$$

donde k es el número de clases en S, k<sub>1</sub> es el número de clases en S<sub>1</sub> y k<sub>2</sub> es el número de clases en S<sub>2</sub>. En caso contrario será rechazado el nuevo punto de corte, y el proceso termina.

#### 4.15 Selección de Atributos

Los clasificadores suelen degradar su comportamiento ante atributos irrelevantes y/o redundantes. La selección de características es un proceso que consiste en seleccionar un subconjunto óptimo de características de una base de datos para reducir su dimensionalidad, eliminar ruido y mejorar el desempeño de un algoritmo de aprendizaje: velocidad de aprendizaje, precisión de la predicción (medido con la tasa de error) y comprensibilidad de los resultados producidos.

El subconjunto óptimo de características está compuesto entonces por las características fuertemente relevantes y las débilmente relevantes pero no redundantes. Encontrar el subconjunto óptimo de características requiere una búsqueda en el espacio de subconjuntos posibles de características de la Base de Datos, que es un problema no determinístico polinomial complejo[Blu92].

Básicamente un proceso de Selección de Características engloba una fase búsqueda y una fase de evaluación del subconjunto resultado de la búsqueda.

Durante la fase de búsqueda se producen subconjuntos de características que son candidatos a ser evaluados. Existen diferentes tipos de búsquedas, pero podemos distinguir tres grandes tipos: la *búsqueda completa* garantiza el hallazgo del subconjunto óptimo, sin tener la necesidad de realizar una búsqueda de todos los posibles subconjuntos (2<sup>n</sup>) del total de n características, que es una búsqueda exhaustiva [Liu05], la *búsqueda secuencial* genera subconjuntos de manera directa, comienza con un subconjunto vacío, para luego agregarle características relevantes de manera progresiva (selección secuencial hacia adelante), o viceversa: comenzar con todo el

conjunto y eliminar características irrelevantes de manera progresiva (selección secuencial hacia atrás) [Liu98] y por último la *búsqueda aleatoria* genera subconjuntos de manera aleatoria, luego aumenta o disminuye características también aleatoriamente para generar el siguiente subconjunto que sería evaluado.

Una vez finalizado el proceso de búsqueda se obtienen subconjuntos de datos que deben ser evaluados. El proceso de evaluación, consiste en medir la optimalidad del subconjunto generado para los fines de un problema de aprendizaje, que en este trabajo es de Clasificación. [Lan94] divide las funciones de evaluación en dos categorías: *Filtro* y *wrapper (envolvente)*. La diferencia entre una función de evaluación tipo filtro o tipo wrapper radica en que en la primera categoría se incluyen los algoritmos en los que la selección de atributos se realiza como un preproceso a la fase clasificación y por tanto de manera independiente, por lo que puede entenderse como un filtrado de los atributos irrelevantes y redundantes. Por otro lado, en los métodos de tipo wrapper [Joh94], la selección de atributos y los algoritmos de aprendizaje no son elementos independientes, ya que se utiliza el comportamiento de un algoritmo de clasificación como criterio de evaluación de los atributos. El modelo wrapper escoge los atributos que demuestran mejor clasificación, ayudando a mejorar el comportamiento del algoritmo de aprendizaje.

En otras palabras el criterio de evaluación *Filtro* es independiente del algoritmo de aprendizaje (por ejemplo Redes Neuronales, Máquinas de Vector Soporte, etc.) mientras que la evaluación tipo *envolvente*, depende del algoritmo de aprendizaje que se use. Por lo tanto el wrapper genera un costo computacional mayor al de los algoritmos pertenecientes al Modelo de Filtro.

Generalmente el proceso de Selección de características se detiene cuando se alcanza el valor de algún parámetro o umbral que se ha establecido o se terminó de realizar la búsqueda completa o se encontró un subconjunto de características óptimo. Los algoritmos de selección de características envolvente se diferencian en el método de búsqueda de la generación del subconjunto. Existe diversos tipos de búsqueda utilizando un algoritmo genético o comenzar con un subconjunto aleatorio de características, buscando subconjuntos que tengan menor número de características y que generen los menores errores de aprendizaje, lo que se llama búsqueda aleatoria[Wit00], y por último la búsqueda “Best First”, el cual realiza la búsqueda en el

espacio de subconjuntos de características atravesando diferentes capas del espacio de búsqueda [Koh97].

Para llevar a cabo nuestro estudio, hemos escogido de cada categoría 2 tipos de métodos con el mismo método de búsqueda utilizado tanto para los de filtro como para los de tipo wrapper. El método de búsqueda escogido es el de Best First, y los tipos de métodos de evaluación utilizados para la categoría filtro han sido CFS (Correlation based Feature Selection) y CNS (Consistency based Feature Selection), mientras que para los de tipo wrapper se escogió un clasificador tipo árbol el C4.5 y el clasificador Naive Bayes, por su rapidez y sencillez. Los algoritmos C4.5 y Naive Bayes han sido explicados en capítulos anteriores.

#### 4.15.1 Selección por Correlation Based Filter “CFS”

Este filtro considera que un buen conjunto de atributos son aquellos que están altamente correlacionados o predictivos con la clase y poco correlacionados entre sí [Hal99]. Trata de encontrar el subconjunto óptimo de atributos altamente correlados con la clase y, al mismo tiempo, con un bajo grado de redundancia entre ellos. Para ello, busca un subconjunto de atributos considerando la capacidad predictora de cada uno individualmente, pero también se busca que haya poca correlación entre los atributos.

La fórmula de la heurística siguiente provee una definición operacional de esta hipótesis:

$$G_s = \frac{k\overline{r_{ci}}}{\sqrt{k \cdot (k - 1)\overline{r_{cii}}}}$$

$k$  es el número de atributos en el subconjunto;  $\overline{r_{ci}}$  es la correlación media con la clase, y  $\overline{r_{cii}}$  es la correlación media de los atributos entre sí. La expresión del heurístico es de hecho el coeficiente de correlación de Pearson [Rod88], donde todas las variables han sido estandarizadas. El numerador puede ser visto como la medida de cuán predictiva de la clase puede ser un subconjunto de atributos dado y el denominador como cuanta redundancia existe, entre los atributos predictores. La bondad de este heurístico es que dejará fuera atributos irrelevantes, ya sean malos predictores o atributos redundantes [Hal99].

#### 4.15.2 Selección Por Consistency Based Filter “CNS”

Básicamente evalúa un subconjunto de atributos por el nivel de consistencia en los valores de la clase al proyectar las instancias de entrenamiento sobre el subconjunto de atributos. Este tipo de métodos buscan combinaciones de atributos cuyos valores dividen los datos en subconjuntos que contienen una gran mayoría de clase única. Por lo general la búsqueda está centrada en pequeños subconjuntos de características con una alta clase de consistencia. Nuestro método basado en la evaluación de la consistencia se basa en el método de Liu y Setiono’s[Liu96]:

$$Consistencia_s = 1 - \frac{\sum_{i=0}^J |D_i| - |M_i|}{N}$$

donde  $s$  es un subconjunto de atributos,  $J$  es el número de combinaciones distintas de valores de un atributo en  $s$ ,  $|D_i|$  es el número de ocurrencias del  $i$ -ésimo combinación de valores del atributo,  $|M_i|$  es la cardinalidad de la clase mayoritaria del  $i$ -ésimo combinación de valores del atributo y  $N$  es el número total de instancias en el conjunto de datos. Un conjunto de datos con atributos numéricos es primero discretizado con el método de Fayyad and Irani [Fay93].





## CAPÍTULO 5 - ESTUDIO EXPERIMENTAL

---

Como se aprecia en [Land94], todos los sistemas informáticos sufren de problemas de seguridad que son técnicamente difícil y económicamente costosos para ser resueltos por los fabricantes. Por tanto el uso de IDS para la detección de ataques es de suma importancia. Sin embargo, cuando revisamos el estado de arte de las soluciones más avanzadas de IDS y de herramientas comerciales, la mayoría de los productos utilizan el método de detección por uso indebido basándose en que la detección de anomalías no es una tecnología madura todavía [Shy03].

Para encontrar la razón de este hecho, vamos a llevar a cabo un estudio detallado enfocado en técnicas basadas en anomalías, examinando diversos aspectos tales como conjunto de datos, selección de atributos, discretización y diferentes técnicas de aprendizaje, creando conjuntos de datos para el entrenamiento y su posterior evaluación (conjuntos de test) y por último someter estos resultados a ANOVA (análisis de la varianza), colección de modelos estadísticos y sus procedimientos asociados para llevar a cabo estudios comparativos.

Los datos utilizados en este estudio son del NSL-KDD Data set [NSL09], el cual es una mejora de los datos del concurso KDD cup'99[Kdd99]. En KDD-99 se utilizó una versión reducida de la amplia variedad de intrusiones militares simuladas en un entorno de red, proporcionadas por DARPA Intrusion Detection Program Evaluation en 1998 [Ken98], que tenían como objetivo evaluar el estudio y la investigación en la detección de intrusiones. Los Laboratorios Lincoln [Mit] crearon un entorno para adquirir un volcado de datos TCP durante nueve semanas, en una red de área local (LAN) que simulaba la típica red de las Fuerzas Aéreas de EE.UU salpicada con múltiples ataques. El conjunto de datos de entrenamiento, obtenidos durante las primeras 7 semanas, ocupaba cerca de cuatro gigabytes, lo que equivale aproximadamente a cinco millones de registros de conexión. Del mismo modo, los datos de test se obtuvieron durante las dos últimas semanas y rondaban dos millones de registros de conexión.

NSL-KDD viene a mejorar los fallos que tiene el conjunto de datos KDD'99. La primera deficiencia importante en el conjunto de datos KDD'99 es el gran número de registros redundantes. El análisis del conjunto de datos train y test, se encontró que alrededor del 78% y 75% de los registros se duplican en el train y en el conjunto de prueba, respectivamente. Esta cantidad grande de registros redundantes en el conjunto de entrenamiento causará que el aprendizaje de los algoritmos estará afectado y con lo cual desviado hacia los registros más frecuentes, y así impedirá el aprender registros menos infrecuentes que son por lo general más dañosos a redes como son los ataques de tipo U2R. Por otra parte, la existencia de estos registros repetidos en el conjunto de prueba, hará que los resultados de evaluación sean influidos por los métodos que tienen mejores tasas de detección sobre los registros frecuentes.

La nueva versión de datos KDD, NSL-KDD está públicamente disponible para investigadores en [NSL09].

Los atributos del NSL-KDD Data se pueden clasificar en 3 grupos:

- 1) **Características básicas:** esta categoría agrupa todos los atributos que se pueden extraer de una conexión TCP / IP. La mayoría de estas características conducen a una demora en la detección.
- 2) **Características del tráfico:** esta categoría incluye las características que se calculan con respecto a un intervalo de la ventana y se divide en dos grupos:
  - Atributos de “mismo host”, que tienen en cuenta sólo las conexiones en los dos últimos segundos que tengan el mismo destino que la conexión actual, y las estadísticas relacionadas con el protocolo, los servicios, etc.
  - Atributos de “mismo servicio”, que examinan sólo las conexiones en los dos últimos segundos que tienen el mismo servicio que la conexión actual.

Los dos tipos de “tráfico” antes mencionados se llaman características basadas en tiempo. Sin embargo, hay varios ataques de sondeo que son lentos y escanean los puertos de los hosts en un intervalo de tiempo mucho mayor que 2 segundos, por ejemplo, uno cada minuto. Como resultado, estos ataques no producen patrones de intrusión en una ventana de tiempo de 2 segundos. Para resolver este problema, las características del “mismo servicio” y del mismo “host” se recalculan basándose en una

ventana de 100 conexiones en lugar de en una ventana de tiempo de 2 segundos. Estas características de conexión se denominan basadas en "tráfico".

**3) Características de contenido:** a diferencia de la mayoría de los ataques de DoS y de sondeo, la R2L y los ataques de U2R no tienen patrones secuenciales frecuentes. Esto es porque el DoS y Probing implican muchas conexiones a algún host (s) en un período muy corto de tiempo, sin embargo ataques R2Ly U2R están incrustados en las porciones de datos de los paquetes, y normalmente sólo implica una única conexión. Para detectar este tipo de ataques, tenemos algunas características para ser capaces de buscar un comportamiento sospechoso en la parte de datos, por ejemplo, el número de intentos de acceso fallidos. Estas características se denominan características de contenido.

Cada registro de conexión está compuesto de 42 atributos, lo que supone unos 100 bytes por registro. Los atributos que componen un único registro se muestran en las siguientes tablas:

Tabla1. Atributos básicos de las conexiones TCP

<b>Atributo</b>	<b>Descripción</b>	<b>Tipo</b>
Duration	Tiempo en segundos de la conexión	Continuo
protocol_type	Tipo de protocolo (TCP, UDP)	Discreto
Service	Tipo de servicio destino (HTTP, Telnet)	Discreto
src_byte	Número de bytes del origen al destino	Discreto
dst_byte	Número de bytes del destino al origen	Discreto
Flag	Estado de la conexión	Categorico
Land	1 si la conexión corresponde mismo/host; 0 de otro modo	Categorico
wrong_fragment	Número de fragmentos "erróneos"	Discreto
Urgent	Número de paquetes urgentes	Discreto

Tabla2. Atributos Derivados de una conexión TCP.

<b>Atributo</b>	<b>Descripción</b>	<b>Tipo</b>
Hot	Número de indicadores "importantes"	Continuo
num_failed_logins	Número de intentos de acceso fallido	Continuo
logged_in	1 acceso exitoso; 0 fallo	Discreto
num_comprised	Número de condiciones sospechosas	Continuo
root_shell	1 si es superusuario; 0 en otro caso	Discreto

su_attempted	1 si se intenta comando “su root”; 0	Discreto
num_root	Número de accesos como root	Continuo
num_file_creations	Número de operaciones de creación de archivos	Continuo
num_shells	Números de Shell prompts abiertos	Continuo
num_access_files	Número de operaciones de control de acceso a archivos	Continuo
num_outbund_cmds	Número de comandos externos (sesión FTP)	Continuo
is_hot_login	1 si login pertenece a la lista “hot”; 0 caso contrario	Discreto
is_guest_login	1 si login es del tipo “guest”, 0 caso contrario	Discreto

Tabla3 Atributos con ventana de 2 segundos.

<b>Atributo</b>	<b>Descripción</b>	<b>Tipo</b>
Count	Número de conexiones a la misma máquina que la conexión actual en los últimos dos segundos	Continuo
	<i>Atributos de conexiones del mismo host</i>	
error_rate	% de conexiones con error “SYN”	Continuo
error_rate	% de conexiones con error “REJ”	Continuo
same_srv_rate	% de conexiones al mismo servicio	Continuo
diff_srv_rate	% de conexiones a diferentes servicios	Continuo
srv_count	Número de conexiones al mismo servicio que la conexión actual en los últimos dos segundos	Continuo
	<i>Atributos de conexiones del mismo servicio</i>	
srv_error_rate	% de conexiones con error “SYN”	Continuo
srv_error_rate	% de conexiones con error “REJ”	Continuo
srv_diffe_host_rate	% de conexiones a diferentes hosts	Continuo

Cabecera de la trama de una conexión del conjunto de datos NSL-Data:

Etiqueta	Atributo
1	duration
2	protocol_type
3	service
4	flag
5	src_bytes
6	dst_bytes
7	land
8	wrong_fragment
9	urgent

Etiqueta	Atributo
10	hot
11	num_failed_logins
12	logged_in
13	num_compromised
14	root_shell
15	su_attempted
16	num_root
17	num_file_creations
18	num_shells
19	num_access_files
20	num_outbound_cmds
21	is_host_login
22	is_guest_login

Etiqueta	Atributo
23	count
24	srv_count
25	serror_rate
26	srv_serror_rate
27	rerror_rate
28	srv_rerror_rate
29	same_srv_rate
30	diff_srv_rate
31	srv_diff_host_rate
32	dst_host_count
33	dst_host_srv_count
34	dst_host_same_srv_rate
35	dst_host_diff_srv_rate
36	dst_host_same_src_port_rate
37	dst_host_srv_diff_host_rate
38	dst_host_serror_rate
39	dst_host_srv_serror_rate
40	dst_host_rerror_rate
41	dst_host_srv_rerror_rate

Ataque	Categoría	Frecuencia
Apache2	Dos	737
Back	Dos	555
Land	Dos	8
Mailbomb	Dos	293
Neptune	Dos	12939
Pod	Dos	79
Processtable	Dos	685
Smurf	Dos	1194
Teardrop	Dos	200
Udpstorm	Dos	2
Normal	Normal	23160
Ipsweep	Probe	851
Miscan	Probe	996
Nmap	Probe	374
Portssweep	Probe	685
Saint	Probe	319
Satan	Probe	1426
Ftp_write	R2L	4
Guess_Passwd	R2L	1241
Imap	R2L	6
Multihop	R2L	20
Named	R2L	17
Phf	R2L	4
Sendmail	R2L	14
Snmpgetattack	R2L	178
Snmpguess	R2L	331
Spy	R2L	1
Warezclient	R2L	181
Warezmaster	R2L	951
Worm	R2L	2
Xlock	R2L	9
Xsnoop	R2L	4
Buffer_Overflow	U2R	26
Httpstunnel	U2R	133
Loadmodule	U2R	3
Perl	U2R	2
Ps	U2R	15
Rootkit	U2R	17
Sqlattack	U2R	2
Xterm	U2R	13

Figura 19 Ataques/categoría/Nº de registros en la Base de datos.

El conjunto de datos [NSL09] contiene un total de 40 tipos de ataques los cuales se clasifican en 5 categorías diferentes (clasificación de Kendall vista en el capítulo 2) como se puede observar en la Fig 19. Para llevar a cabo el estudio analizaremos este conjunto de datos de 3 formas distintas.

Como se puede observar en la Fig 19, la cantidad de ataques presentes en el conjunto de datos global y de manera general los registros de tipos Probe, R2L y U2R, especialmente estos últimos, quedan en desventaja frente a los registros de tipo Normal y Dos.

En el primer estudio el más sencillo, se pretenderá analizar determinados algoritmos de aprendizaje para el modelado del sistema, a nivel de detección de si una conexión es de tipo ataque -independientemente del tipo de ataque que sea- o por el contrario es de tipo normal -situación de no amenaza-. Por tanto el análisis es a nivel de 2 categorías: “ataque” y “normal”. Se crearán ficheros **balanceados** -igual cantidad de ataques y normal en el fichero de entrenamiento-, para no favorecer el sobreentrenamiento por parte del modelo favoreciendo a una categoría con más presencia de registros, donde el conjunto de train contendrá un 70% de conexiones y el de test el 30% restante.

En el segundo caso de estudio, la situación se complica un poco más, los datos son clasificados en 5 categorías –este es el método más utilizado en la literatura- y se crearán dos ficheros uno de entrenamiento y otro de test con el 70 y el 30% de datos, en esta situación los datos no están balanceados –simulando una situación más real- y estudiaremos el comportamiento de todos los algoritmos explicados en el capítulo 4 a la hora de clasificar los ataques en 5 categorías.

En el tercer estudio se procederá de la misma manera que en el primer caso creando ficheros balanceados sólo que el estudio se realizara a un nivel más complejo, ya no se pretenderá detectar la categoría de un ataque sino un ataque en concreto.

Hemos explicado la perspectiva por la cual abordaremos el estudio de los sistemas de detección de intrusos basados en anomalías, los cuales utilizan algoritmos de aprendizaje automático para su motor de análisis, para clasificar entre situación de ataque o situación de normalidad.

En estos 3 estudios se procederá a la discretización de los conjuntos de datos mediante dos técnicas distintas. Una de ellas es la de Fayyad & Irani y la otra técnica es la de

Intervalos de igual Frecuencia, que como se explico en el apartado 4.13, éste método, suponiendo que el atributo a discretizar tiene  $m$  valores distintos, este discretizador divide el dominio de cada variable en  $n$  partes, donde cada parte tiene  $m/n$  valores continuos del atributo, siendo el valor escogido para  $n$ , 100.

A parte de la discretización se llevo a cabo un proceso de selección de atributos utilizando tanto métodos de tipo filter como de tipo wrapper. Para el tipo filtro se escogieron un método CFS y otro basado en consistencia, CNS, explicados anteriormente en el capítulo 4. Para el tipo wrapper se seleccionaron, un clasificado envolvente basado en árboles de decisión C4.5 y Naive Bayes. En todos los casos el método de búsqueda escogido ha sido el Best First.

En la tabla Fig.20 se puede encontrar que de los 41 atributos que compone el registro de una conexión cuales son los que han sido seleccionados por estos métodos.

En los 3 casos tendremos un conjunto de datos sin discretizar con el total de atributos, un conjunto de datos discretizado con el método de Fayyad e Irani , y otro con una discretización no supervisada de Intervalos de igual frecuencia.

Adicionalmente se crearan nuevos conjuntos de datos basados en los conjuntos anteriores utilizando solamente los atributos ofrecidos por los métodos de selección de características, como puede observarse en las Figuras 20,21,22, 23 y 24.

En resumen tendremos conjuntos de datos sin discretizar y sin selección de atributos, conjuntos de datos discretizados y sin selección de atributos y una nueva combinación de conjuntos de datos: sin discretizar aplicándoles los 4 métodos de selección de atributos, otra de conjuntos de datos discretizados con la técnica de Fayyad & Irani y con la aplicación de la selección de atributos y por último conjuntos discretizados por la técnica de intervalos de igual frecuencia ( $n=100$  en todos los casos) y combinados con los 4 métodos de selección de atributos.

Finalmente dispondremos para cada caso de estudio un total de 15 ficheros train y 15 ficheros para la fase de test.



Método	Atributos	Total
CFS	3, 4, 5, 6, 11, 14, 25, 29, 30, 37	10
CNS	1, 3, 5, 6, 12, 23, 24, 29, 30, 33, 34, 35, 36, 38, 40	15
C4.5	1, 2, 3, 4, 6, 8, 10, 11, 12, 14, 15, 18, 19, 22, 23, 27, 29, 30, 32, 33, 34, 35, 36, 37, 38	25
Naive Bayes	2, 3, 4, 8, 9, 11, 18, 21, 22, 30, 32	11

Figura 20 Atributos Seleccionados por los distintos métodos

Etiqueta	Filtro CFS
3	service
4	flag
5	src_bytes
6	dst_bytes
11	num_failed_logins
14	root_shell
25	error_rate
29	same_srv_rate
30	diff_srv_rate
37	dst_host_srv_diff_host_rate

Figura 21 Selección Atributos Filtro CFS

Etiqueta	Filtro CNS
1	duration
3	service
5	src_bytes
6	dst_bytes
12	logged_in
23	count
24	srv_count
29	same_srv_rate
30	diff_srv_rate
33	dst_host_srv_count
34	dst_host_same_srv_rate
35	dst_host_diff_srv_rate
36	dst_host_same_src_port_rate
38	dst_host_error_rate
40	dst_host_error_rate

Figura 22 Atributos Filtro CNS

Etiqueta	C4.5
1	duration
2	protocol_type
3	service
4	flag
6	dst_bytes
8	wrong_fragment
10	hot
11	num_failed_logins
12	logged_in
14	root_shell
15	su_attempted
18	num_shells
19	num_access_files
22	is_guest_login
23	count
27	error_rate
29	same_srv_rate
30	diff_srv_rate
32	dst_host_count
33	dst_host_srv_count
34	dst_host_same_srv_rate
35	dst_host_diff_srv_rate
36	dst_host_same_src_port_rate
37	dst_host_srv_diff_host_rate
38	dst_host_error_rate

Figura 23 Atributos Wrapper C4.5

Etiqueta	NB
2	protocol_type
3	service
4	flag
8	wrong_fragment
9	urgent
11	num_failed_logins
18	num_shells
21	is_host_login
22	is_guest_login
30	diff_srv_rate
32	dst_host_count

Figura 24 Atributos Wrapper Naive Bayes

## 5.1 Primer estudio a nivel 2 Categorías: “Normal” y “Ataque”

Como se comentó en párrafos superiores el fichero de entrenamiento no es un fichero balanceado, esto supondrá que en la fase de entrenamiento del modelo ciertos algoritmos favorecerán más a una clase que a otra.

En esta parte del estudio, el estudio es a nivel de detección de si una conexión es del tipo ataque o del tipo normal. El sistema se entrena con un conjunto de datos que solamente contienen dos tipos de registros, “ataque” o “normal”. Para la construcción de un conjunto de datos balanceado, la selección de los ataques se realizó de la siguiente manera: de cada categoría (Dos, Probe, R2L y U2R Fig19) se seleccionaron aleatoriamente 252 ataques. Este número se debe a que la suma total de los ataques de tipo U2R es de 252. Al final se obtuvieron 252 ataques \* 4 categorías = 1008 ataques. De las conexiones de tipo normal se seleccionaron aleatoriamente esta misma cantidad 1008. En base a esto, se construyó un fichero de 2016 registros que contiene 2 tipos de conexiones ataque y normal, y a partir de él se obtuvo un fichero de entrenamiento con un 70% de datos y un fichero de test con el 30% restante.

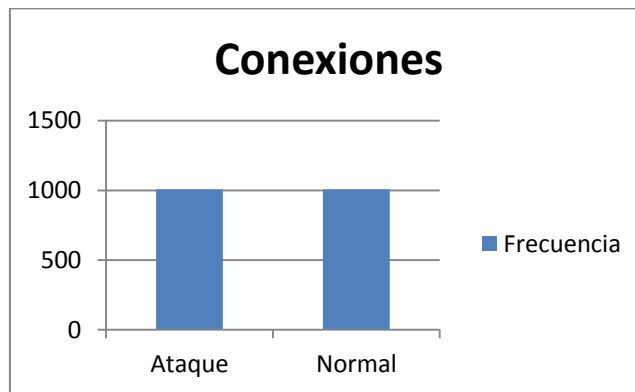


Figura 25 Conjunto balanceado con 2 Categorías

Utilizando la discretización, el conjunto de datos se reduce y se construyeron los ficheros de entrenamiento y test de tal forma que estuviesen equilibrados, como se puede observar en las siguientes tablas:

Tipo	Train	Test
Ataque	706	302
Normal	706	302

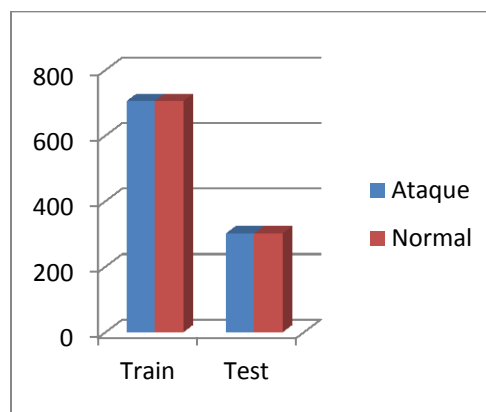


Figura 26 Conjunto Sin Discretizar

Tipo	Train	Test
Ataque	320	253
Normal	320	138

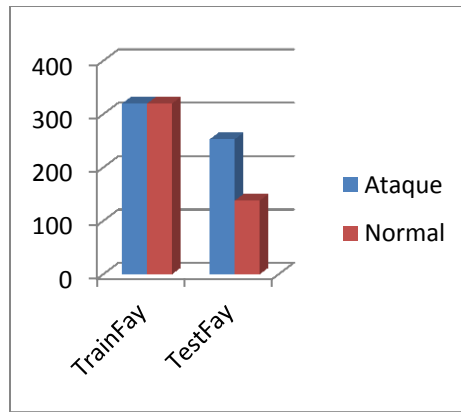


Figura 27 Discretización Fayyad & Irani

Tipo	Train	Test
Ataque	698	275
Normal	698	298

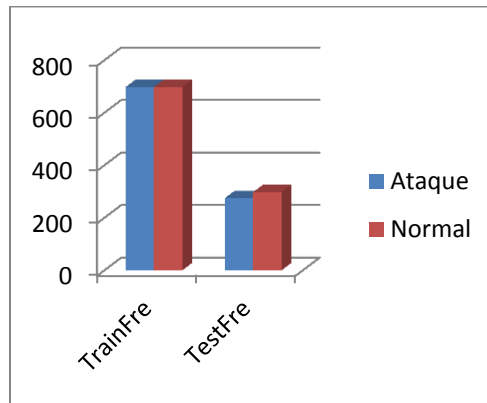


Figura 28 Discretización Intervalo Igual de Frecuencias (n = 100)

## 5.2 Primer estudio a nivel de 5 Categorías: “Dos”, “Probe”, “R2L” y “U2R”.

Como puede observarse en la Fig29 podemos ver la distribución y frecuencia de los ataques en el conjunto de datos, pudiéndose observar que hay ciertos tipos de ataques que prevalecen sobre otros:

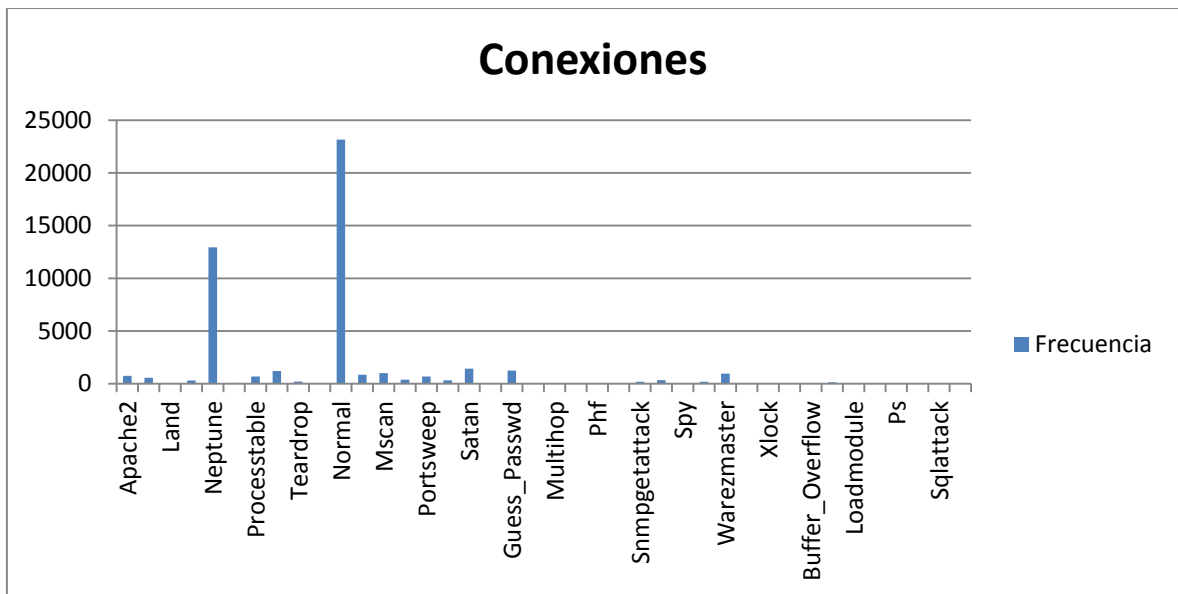


Figura 29 Presencia de ataques en el conjunto de datos

Estos mismos datos pueden ser observados por categorías:

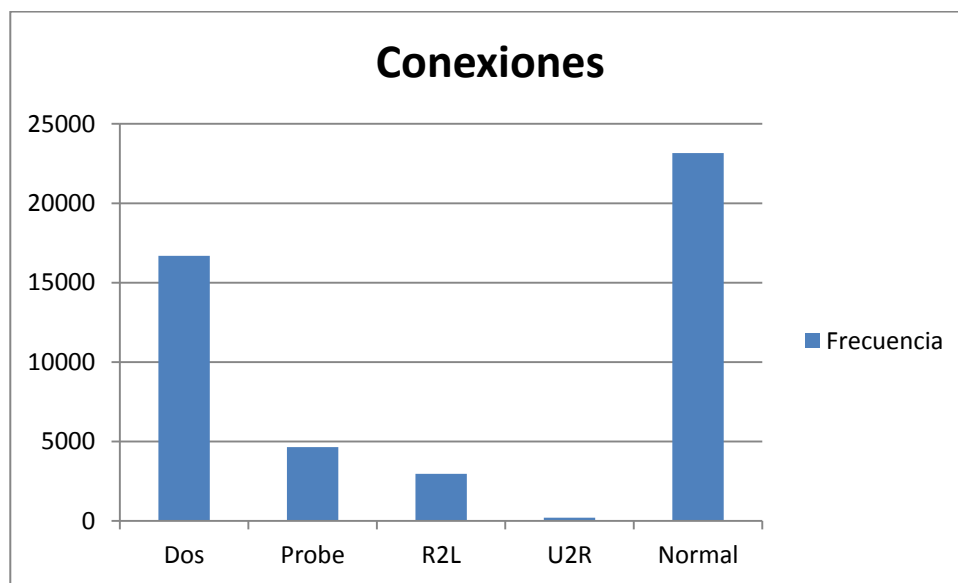


Figura 30 Presencia de ataques en el conjunto de datos por categorías

Para este caso de estudio los ataques son “mapeados” a la categoría a la que corresponden y el sistema es entrenado con registros de tipo “normal”, “Dos”, “Probe”, “R2L” o “U2R”. Esta es la clasificación más estudiada en la literatura.

Categoria	Train	Test
Normal	20.829	2315
Dos	14230	2416
Probe	3282	1410
R2L	2076	892
U2R	147	65

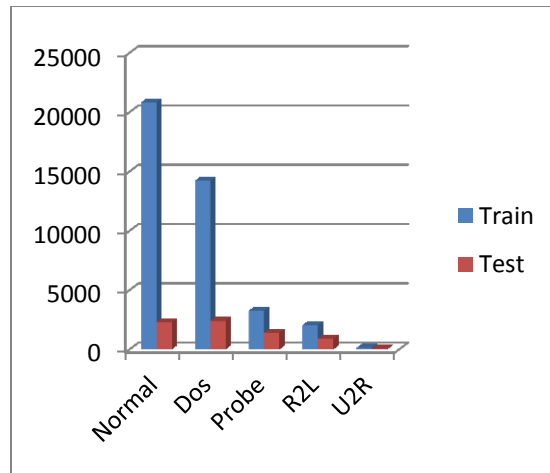


Figura 31 Conjunto de datos Sin Discretizar

Categoria	Train	Test
Normal	10.099	4328
Dos	4525	1939
Probe	1243	533
R2L	785	336
U2R	78	29

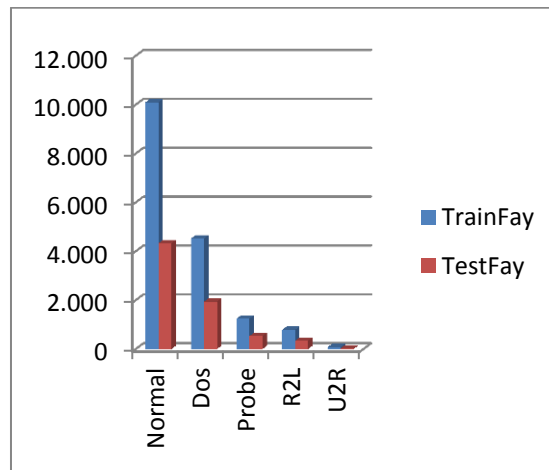


Figura 32 Conjunto de datos Discretizacion Fayyad & Irani

Categoria	Train	Test
Normal	13.938	5973
Dos	8824	3782
Probe	1707	732
R2L	1075	461
U2R	99	36

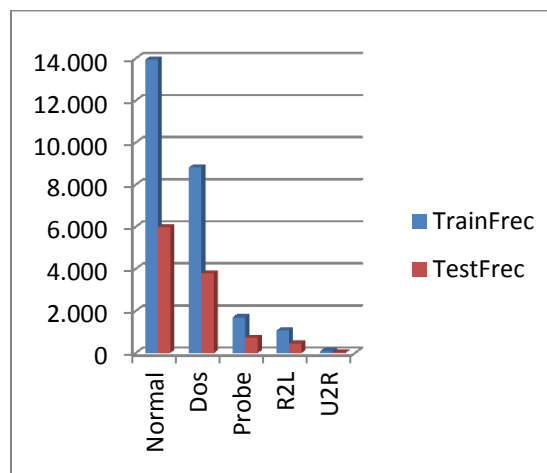


Figura 33 Conjunto de datos Discretizacion Intervalo Igual de Frecuencias (n = 100)

Cómo ocurría en el caso anterior, la discretización reduce considerablemente el conjunto de datos ver Figuras 32 y 33.

### 5.3 Tercer Estudio: A Nivel de Ataque.

En esta parte, el estudio se realizó a nivel de ataque y no de categorías. Se seleccionaron 133 ataques de cada categoría. Se escogió esta cantidad porque de la categoría U2R el ataque con más registros es el de tipo “httptunnel” con una presencia de 133 entradas.

Ataque	Categoría	Cantidad
Normal	Normal	23160
Neptune	DOS	12939
Smurf	DOS	1194
Apache2	DOS	737
Processtable	DOS	685
Back	DOS	555
Mailbomb	DOS	293
Teardrop	DOS	200
Satan	Probe	1426
Miscan	Probe	996
Ipsweep	Probe	851
PortswEEP	Probe	685
Nmap	Probe	374
Saint	Probe	319
Guess_Passwd	R2L	1241
WareZmaster	R2L	951
SnmPguess	R2L	331
WareZclient	R2L	181
SnmPgetattack	R2L	178
Httpstunnel	U2R	133

Figura 34 Elección del Umbral httptunnel = 133

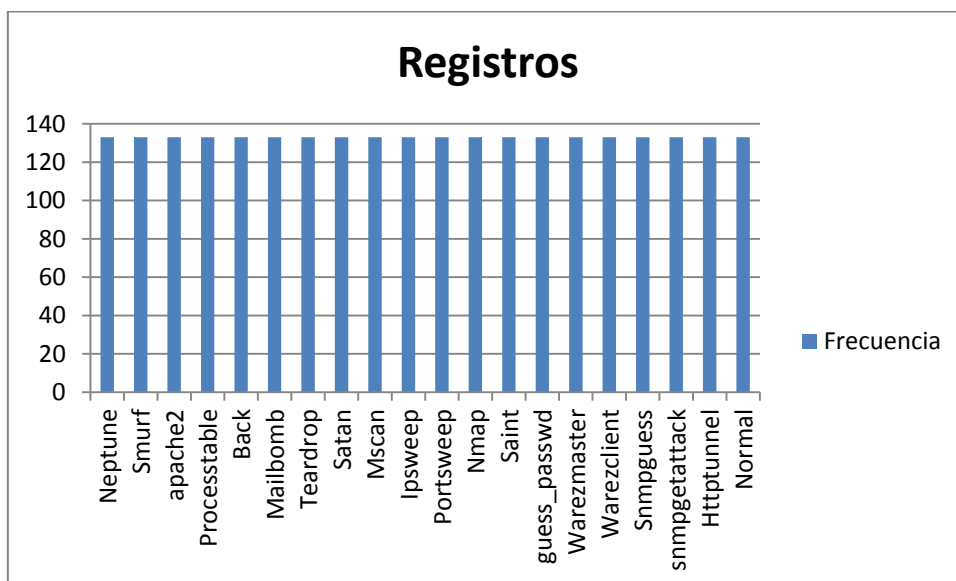


Figura 35Conjunto de ataques balanceado

Nuestro objetivo es el de crear un conjunto de datos balanceado independientemente del método de discretización que se quiera utilizar. Tras la discretización tanto en Fayyad como en la de igual intervalos de frecuencia nuestro conjunto de datos resultado constituía en las siguientes tablas:

Ataque	Categoría	Fayyad
Mailbomb	DOS	41
Ipsweep	Probe	41
Processtable	DOS	42
Snmpguess	R2L	45
Saint	Probe	49
Smurf	DOS	55
Nmap	Probe	55
Httpptunnel	U2R	57
Teardrop	DOS	71
Warezclient	R2L	71
Apache2	DOS	72
Neptune	DOS	78
Portswweep	Probe	79
Snmpgetattack	R2L	80
Guess_Passwd	R2L	87
Warezmaster	R2L	88
Satan	Probe	92
Back	DOS	100
Normal	Normal	123
Mscan	Probe	128

Figura 36Fayyad

Ataque	Categoría	Frec.
Smurf	DOS	103
Processtable	DOS	108
Portswweep	Probe	111
Mailbomb	DOS	113
Nmap	Probe	116
Snmpguess	R2L	118
Snmpgetattack	R2L	119
Guess_Passwd	R2L	121
Httpptunnel	U2R	121
Saint	Probe	122
Satan	Probe	124
Apache2	DOS	125
Teardrop	DOS	125
Warezclient	R2L	128
Ipsweep	Probe	130
Back	DOS	131
Warezmaster	R2L	132
Normal	Normal	133
Neptune	DOS	133
Mscan	Probe	133

Figura 37Intervalo Igual Frec.

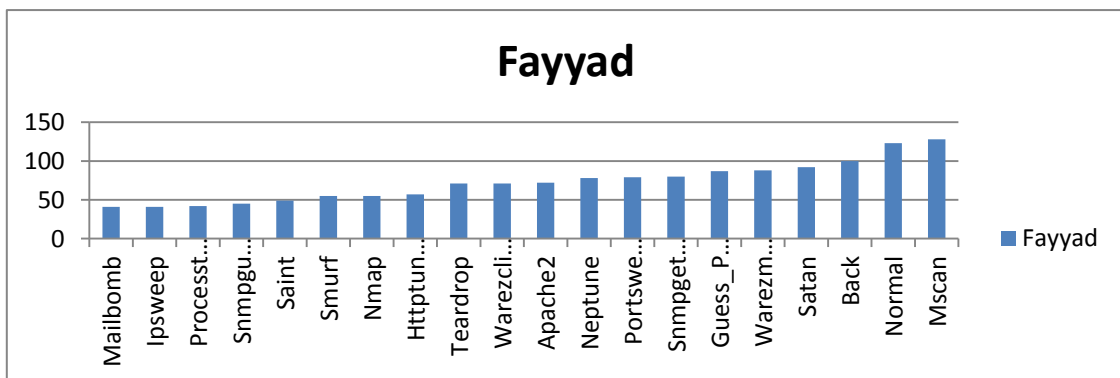


Figura 38Representación gráfica del resultado tras aplicar Discretización de Fayyad & Irani

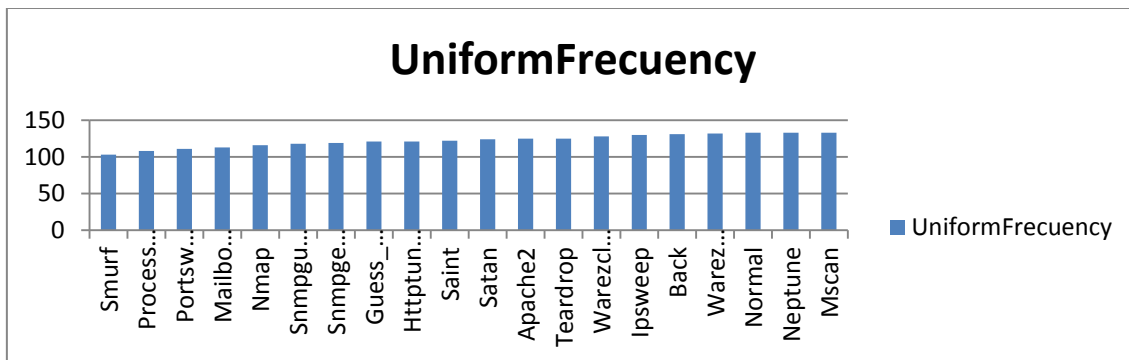


Figura 39 Representación gráfica del resultado tras aplicar Discretización Igual Intervalo de Frec.

Para realizar los experimentos con los nuevos datos discretizados se ha optado por crear nuevos conjuntos balanceados, tomando como regla, el escoger como umbral una cantidad de ataques igual al ataque con menor presencia en el conjunto, es decir, en el conjunto de Fayyad se seleccionaron 41 registro de cada ataque y en el conjunto de intervalos de frecuencias uniformes se seleccionaron 103 registros.

Ataque	Train	Test
Normal	94	39
Neptune	94	39
Smurf	94	39
Apache2	94	39
Processtable	94	39
Back	94	39
Mailbomb	94	39
Teardrop	94	39
Satan	94	39
Mscan	94	39
Ipsweep	94	39
PortswEEP	94	39
Nmap	94	39
Saint	94	39
Guess_Passwd	94	39
Warezmaster	94	39
Snmpguess	94	39
Warezclient	94	39
Snmpgetattack	94	39
Httptunnel	94	39

Figura 40 Sin Discretizar

Ataque	Train	Test
Normal	29	12
Neptune	29	12
Smurf	29	12
Apache2	29	12
Processtable	29	12
Back	29	12
Mailbomb	29	12
Teardrop	29	12
Satan	29	12
Mscan	29	12
Ipsweep	29	12
PortswEEP	29	12
Nmap	29	12
Saint	29	12
Guess_Passwd	29	12
Warezmaster	29	12
Snmpguess	29	12
Warezclient	29	12
Snmpgetattack	29	12
Httptunnel	29	12

Figura 41 Fayyad

Ataque	Train	Test
Normal	73	30
Neptune	73	30
Smurf	73	30
Apache2	73	30
Processtable	73	30
Back	73	30
Mailbomb	73	30
Teardrop	73	30
Satan	73	30
Mscan	73	30
Ipsweep	73	30
PortswEEP	73	30
Nmap	73	30
Saint	73	30
Guess_Passwd	73	30
Warezmaster	73	30
Snmpguess	73	30
Warezclient	73	30
Snmpgetattack	73	30
Httptunnel	73	30

Figura 42 Intervalo Igual Frec.

## 5.4 Construcción de Modelos.

En el capítulo 4, se explicaron los algoritmos seleccionados para cada técnica de aprendizaje para construir un clasificador. Todos los conjuntos de datos obtenidos



fueron sometidos a diferentes técnicas de construcción de un sistema basado en anomalías. Los algoritmos de aprendizaje utilizados han sido:

Paradigma	Algoritmo	Variante
Inmune	Clonalg	
Genético	Programación Genética	
Fuzzy	Fuzzy Unordered Rule Induction Algorithm (Furia)	
Inducción Reglas	Repeated Incremental Pruning to Produce Error Reduction (RIPPER)	
Inducción Reglas	Partial Decision Trees (Part)	
Bayesiano	Naive Bayes	
Red Bayesiana	Tree Augmented Naive Bayes (Tan)	
Vecino más cercano	KNN-1	K = 1
Vecino más cercano	KNN-50	K = 50
SVM	SMD	Polykernel
SVM	SMD	Gaussiano
SBM	CSV	Gaussiano
SVM	CSV	Sigmoide
Red Neuronal	Perceptrón Multicapa	
Red de Base Radial	RBF	K = 5
Árboles de Decisión	C4.5	
Árboles de Decisión	Random Forest	10 árboles y 6 atributos aleatorios
Árboles de Decisión	NBTREE	
Árboles de Decisión	CART	
Modelos Ocultos de Markov	Discreto	Diferente Nº de Estados 11,15,25

Figura 43 Algoritmos aplicados.

En el caso de los modelos ocultos de Markov al tratarse de modelos discretos en los 3 casos sólo se trabajó con los conjuntos de datos discretizados sin selección de atributos y los discretizados aplicándoles selección de atributos. Debido a que el número de estados ocultos pueden afectar al rendimiento de la clasificación, se utilizan HMM con diferentes números de estados ocultos para hacer la clasificación. El número de estados elegido han sido 11, 15 y 25 basándonos en el número de atributos que los distintos métodos de selección de características han dado como resultado.

Por último tanto para el primer caso estudio de 2 categorías (Ataque/Normal) y el último caso basado en ataques, solamente se seleccionaron los siguientes algoritmos: FURIA, TAN, KNN-1, SMO (polykernel), C4.5, RANDOM FOREST y MODELOS OCULTOS de MARKOV, debido a los buenos resultados que han dado.

La evaluación de un experimento de reconocimiento de patrones se basa en la medida del acierto (dado como porcentaje de muestras o instancias bien clasificadas) de un conjunto de datos, también llamado conjunto de test. Por tanto de este estudio se

obtuvieron los tiempos de construcción de los modelos y las matrices de confusión (que se pueden consultar en el apéndice A de este trabajo), así como la tasa de acierto Global de cada clasificador y la tasa de acierto de:

- en el caso del primer estudio Tasa de Acierto: Ataque y Normal.
- en el caso del segundo estudio tasa de Acierto: Dos, Probe, R2L, U2R y normal.
- en el caso del tercer estudio Tasa de Acierto de cada uno de los ataques de la Fig40.

Cabe resaltar que en el primer caso de dos categorías se llevaron a cabo 120 ejecuciones, en el segundo caso de 5 categorías se hicieron 315 ejecuciones y en el último caso otras 120 ejecuciones lo que hace un total de 555 ejecuciones. En el apéndice A se puede consultar la tabla de tiempos y aciertos obtenidos para cada caso y en el apéndice B se pueden consultar las matrices de confusión de todas las ejecuciones llevadas a cabo.

Basándonos en estos datos se procederá a su análisis estadístico.



## CAPÍTULO 6 - ANÁLISIS ESTADÍSTICO ANOVA

---

### 6.1 Estudio estadístico de factores en detección de intrusos.

Para la aplicación del análisis de la varianza, las observaciones de las variables respuesta se expresan como el resultado aditivo de una serie de componentes. En general, si tenemos múltiples factores, la observación  $Y_{a,b \dots l,s}$ , donde:

- $a, b \dots l$  son las diferentes variables o factores del problema
- $s$  es el número de observaciones cuando existe repetición de experimentos reales bajo las mismas condiciones o valores de los factores.

Puede ser admitida, en una primera aproximación, que es consecuencia aditiva de los efectos de los factores  $a, \dots, l$ . Por tanto la observación puede ser expresada como una suma lineal:

$$Y_{a,b,\dots,l,s} = a + b + \dots + l + \varepsilon$$

Siendo  $\varepsilon$  el efecto llamado de causas no asignables o de azar. Si los factores  $a, b, \dots, l$  se mantienen constantes, las medidas de  $Y_{a,b \dots l,s}$  presentarán variaciones que deberán poder considerarse como atribuibles a un gran número de pequeñas causas indistinguibles entre sí, que es lo que se designa como variación aleatoria. Si disponemos un experimento en que midamos  $Y_{a,b \dots l,s}$  a niveles diferentes de uno o más factores, el conjunto de medidas obtenidas puede que no sea homogéneo, estando formado por dos o más grupos. El propósito de la técnica introducida por Fisher es precisamente contrastar esta heterogeneidad, para ver si tales factores son realmente causas asignables en la variación que se trata de estudiar, o bien, se debe atribuir dicha variación al efecto de azar. Es decir, en el análisis de la varianza se trata de separar las componentes de la variación que aparecen en un conjunto de datos estadísticos, determinando si la discrepancia entre las medias de los factores son mayores de lo que podría esperarse razonablemente de las variaciones que ocurren dentro de los factores. De forma más precisa, en el análisis de la varianza una observación es el resultado aditivo de los siguientes componentes:

Un **efecto común** para el conjunto de todas las observaciones del experimento, denominado efecto fijo o común.

Un **efecto específico** debido a la presencia del nivel concreto de cada factor considerado como variable de entrada (se denomina efecto principal).

Un **efecto combinado** debido a la presencia de los niveles concretos de dos o más variables presentes en el experimento, es lo que se denominan interacciones entre los efectos principales. Al número de factores principales que están involucrados en la interacción se le denomina orden de interacción.

Un **residuo o error aleatorio** que corresponde a la desviación entre lo realmente observado experimentalmente y lo ajustado por el modelo estadístico.

En una primera etapa, mediante el análisis de la varianza se determina si la hipótesis nula es cierta o no, es decir, si todos los efectos de los distintos niveles de un determinado factor son iguales entre sí o bien si todas las interacciones de un cierto orden son nulas. Con esto se pretende constatar qué factores producen alteraciones significativas de la variable respuesta, al cambiar el nivel en que es subdividido el dominio de los posibles valores que dicho factor puede tomar. En caso de que la hipótesis nula fuese rechazada, se suele realizar un estudio más profundo encaminado a clasificar los niveles de los factores más significativos, en función de la magnitud de su efecto principal, y detectar diferencias sobre la variable respuesta por el uso de un determinado nivel. Con ello podemos concluir qué bloque funcional del proceso de detección de intrusos tiene una mayor repercusión en la obtención de la salida del sistema, al cambiar el diseño de dicha función. Una de la mayor fuente de información es la tabla ANOVA

Comenzamos a estudiar diversos ejemplos, donde vamos a tratar de analizar estadísticamente como influye la selección de diversos valores o niveles en los efectos principales:

1. Tipo de Filtro
2. Tipo de Discretización
3. Tipo de Algoritmo

En dos variables de salida:

1. El error global del sistema en la clasificación de intrusos
2. El tiempo de cómputo necesario.

Para ello, se van a realizar tres grandes grupos de experimentos:

1. Experimentos donde solo existen dos categorías en la variable de salida de clasificación: Ataque y Normal
2. Experimentos donde existen cinco categorías en la variable de salida de clasificación: cuatro serán para clasificar diferentes ataques: Dos, Probe, R2L, U2R y finalmente cuando la salida se puede considerar Normal
3. Experimentos donde existen una gran cantidad de categorías para la variable de salida de clasificación: Normal, Neptune, Smurf, Apache2, Processtable, Back, Mailbomb, teardrop, Satan, Mscan, Ipsweep, Portsweep, Nmap, Saint, Guess\_passwd, Warezmaster, Warezclient, Smpguess, Smpgetattack, httptunnel.

## 6.2 Experimentos donde sólo existen dos categorías en la variable de salida de clasificación: Ataque y Normal

Como se ha comentado anteriormente, llevaremos a cabo el estudio estadístico de dos variables dependientes a las que queremos analizar:

1. El error global del sistema en la clasificación de intrusos
2. El tiempo de computo necesario

### 6.2.1 Análisis estadístico del error global con dos categorías en la variable de salida

Vamos a comenzar analizando el error global del sistema en la clasificación de intrusos, que vamos a denominar como variable dependiente: AcGlobal, siendo nuestros tres factores o efectos principales:

1. Tipo de Filtro (Filtro) : Aplicación de selección de atributos.
2. Tipo de Discretización (Discr): Aplicación de la Discretización.
3. Tipo de Algoritmo (ALG)

En primer lugar, se va a realizar el estudio de la tabla ANOVA. Este procedimiento ejecuta un análisis de varianza de varios factores para AcGlobal.

#### Análisis de Varianza para AcGlobal - Suma de Cuadrados Tipo III

<i>Fuente</i>	<i>Suma de Cuadrados</i>	<i>Gl</i>	<i>Cuadrado Medio</i>	<i>Razón-F</i>	<i>Valor-P</i>
EFFECTOS PRINCIPALES					
A:Filtro	279,592	4	69,8981	4,81	0,0013
B:Discr	397,317	2	198,658	13,66	0,0000
C:ALG	2076,65	8	259,581	17,86	0,0000
RESIDUOS	1526,5	105	14,5381		
TOTAL (CORREGIDO)	4720,13	119			

Todas las razones-F se basan en el cuadrado medio del error residual

La tabla ANOVA descompone la variabilidad de AcGlobal en contribuciones debidas a varios factores. Puesto que se ha escogido la suma de cuadrados Tipo III (por omisión), la contribución de cada factor se mide eliminando los efectos de los demás factores. Los valores-P prueban la significancia estadística de cada uno de los factores. Puesto que 3 valores-P son menores que 0,05, estos factores tienen un efecto estadísticamente significativo sobre AcGlobal con un 95,0% de nivel de confianza.

Por tanto, como primera conclusión, se puede decir que los tres efectos principales tienen una repercusión relevante sobre la precisión global del sistema de clasificación de detección de intrusos. Aunque los tres valores de Valor-P son menores que 0.05, lo que ahora es interesante de analizar, es si existen diferentes valores de Filtro, o de Discretización o de Algoritmo, que se comporten de forma similar, y por lo tanto, intentar obtener grupos homogéneos o heterogéneos de estos tres efectos o factores. Para ello vamos a proceder a realizar la prueba de múltiples rangos.

Con la realización de estas tablas de múltiples rangos, se realizan comparaciones múltiples para determinar cuáles medias son significativamente diferentes de otras. Las que sean homogéneas o similares desde el punto de vista estadístico, tienen una X en la misma columna, y por tanto se pueden considerar como un grupo homogéneo. Si existen X en diversas columnas, significa que existen más de un grupo. Esto conlleva a que por tanto este factor sea estadísticamente significativo.

### 6.2.1.1 Estudio de tablas de múltiple rangos para ACGlobal

Comenzamos estudiando el factor Filtro, siendo la tabla de rangos múltiples la siguiente:

<i>Filtro</i>	<i>Casos</i>	<i>Media LS</i>	<i>Sigma LS</i>	<i>Grupos Homogéneos</i>
fnb	24	89,3261	0,786901	×
fcns	24	92,3124	0,786901	×
all	24	92,8357	0,786901	×
fc45	24	92,9961	0,786901	×
fcfs	24	93,7324	0,786901	×

<i>Contraste</i>	<i>Sig.</i>	<i>Diferencia</i>	<i>+/- Límites</i>
all - fc45		-0,160417	2,18246
all - fcfs		-0,896667	2,18246
all - fcns		0,523333	2,18246
all - fnb	*	3,50958	2,18246
fc45 - fcfs		-0,73625	2,18246
fc45 - fcns		0,68375	2,18246
fc45 - fnb	*	3,67	2,18246
fcfs - fcns		1,42	2,18246
fcfs - fnb	*	4,40625	2,18246
fcns - fnb	*	2,98625	2,18246

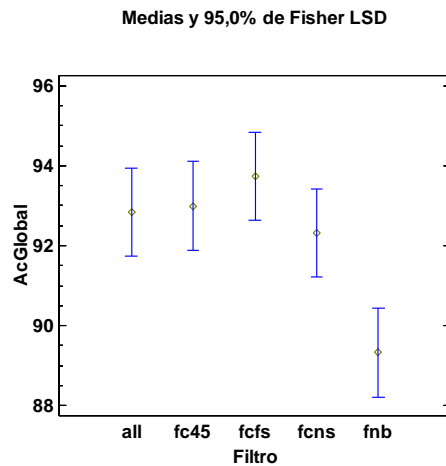
\* indica una diferencia significativa.

Recordemos que en la parte experimental sometimos a los datos a dos tipos de selección de características una de tipo Filtro y otra de tipo wrapper. FNB (naive bayes) y FC4.5 (árbol de decisión) se corresponden a los de tipo wrapper mientras que FCFS y FCNS a los de tipo filtro, y por último definimos ALL como la ausencia de aplicación de la selección de atributos puesto que todos los atributos son utilizados

En la tabla de rangos múltiples para el factor filtro, se puede observar que hay dos grupos: un primer grupo que tendría al tipo de filtro FNB, que sería el que peor valor de precisión obtendría, ya que la media es de 89,3261 % de clasificación, y un segundo grupo homogéneo, donde tendría los restantes tipos de filtros: FCNS, ALL, FC4.5, FCFS. Es decir, desde el punto de vista estadístico, y para la variable de salida de precisión global, estos cuatro tipos de filtros son equivalentes o de forma similar, tienen el mismo comportamiento.

De forma gráfica, esta tabla de rangos múltiples se puede visualizar de la forma:





Continuando, analizamos el estudio del factor **Discr**, siendo la tabla de rangos múltiples la siguiente:

Método: 95,0 porcentaje LSD

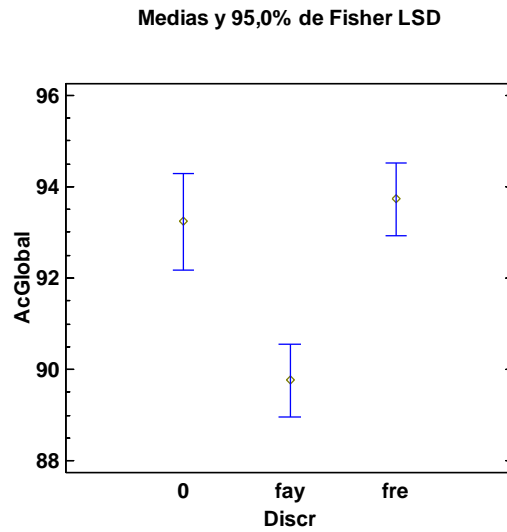
<i>Discr</i>	<i>Casos</i>	<i>Media LS</i>	<i>Sigma LS</i>	<i>Grupos Homogéneos</i>
Fay	45	89,7642	0,568391	x
0	30	93,2317	0,75191	x
Fre	45	93,7258	0,568391	x

<i>Contraste</i>	<i>Sig.</i>	<i>Diferencia</i>	<i>+/- Límites</i>
0 - fay	*	3,46744	1,86894
0 - fre		-0,494111	1,86894
fay - fre	*	-3,96156	1,59384

\* indica una diferencia significativa.

En este caso, para la variable Discretización, existen dos grupos homogéneos, entre los que no existe intersección entre ellos. El primer grupo lo compone el tipo de discretización denominada: FAY, método supervisado de Fayyad & Irani que tiene, desde el punto de vista estadístico, el peor comportamiento. El segundo grupo lo compone los métodos denominados: 0 -no se utiliza discretización- y FRE - discretización no supervisada de Intervalos de igual frecuencia- donde el método FRE es el que mejor valor en media tiene en la precisión de la clasificación.

De forma gráfica, esta tabla de rangos múltiples se puede visualizar de la forma:



Finalmente, realizamos el test de rangos múltiples para el factor tipo de Algoritmo, obteniendo:

Método: 95,0 porcentaje LSD

<i>ALG</i>	<i>Casos</i>	<i>Media LS</i>	<i>Sigma LS</i>	<i>Grupos Homogéneos</i>
Markov15	10	85,1596	1,23878	X
Markov25	10	86,2586	1,23878	X
Markov11	10	86,3876	1,23878	X
SmoPoly	15	93,2673	0,984482	X
Furia	15	93,9353	0,984482	X
TAN	15	94,5847	0,984482	X
C4,5	15	95,062	0,984482	XX
RandomForest	15	97,748	0,984482	X
KNN-1	15	97,762	0,984482	X

Como se puede ver en esta tabla, existen tres grupos diferentes en los que se pueden agrupar los algoritmos utilizados, mencionando que no existen diferencias estadísticamente significativas entre aquellos niveles que compartan una misma columna de X's.

En el primer grupo comentar que estarían las diferentes variantes de Modelos de Markov: MARKOV15, MARKOV25 y MARKOV11, siendo este grupo el que peor resultados obtiene. Existe un segundo grupo con los algoritmos: SMO-Poly, FURIA, TAN y C4.5. Y finalmente existe el grupo tercero, con intersección con el segundo con

el algoritmo C4.5, que estaría formado por los métodos: C4.5, RANDOM FOREST y KNN-1. Como conclusión importante, indicar que este grupo es el que mejor resultados obtiene, siendo el KNN-1 el mejor algoritmo.

De forma detallada, podemos ver entre que diferentes tipos de algoritmos existen diferencias que pueden ser consideradas estadísticamente significativas, y que por tanto son marcadas con un asterisco en la siguiente tabla:

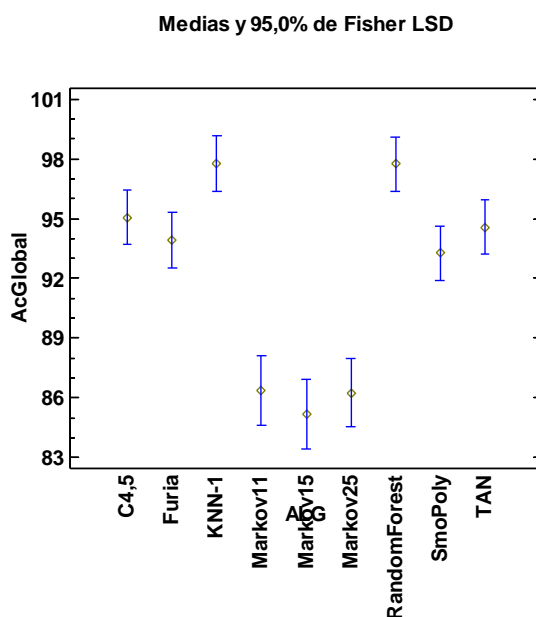
<i>Contraste</i>	<i>Sig.</i>	<i>Diferencia</i>	<i>+/- Límites</i>
C4,5 - Furia		1,12667	2,76061
C4,5 - KNN-1		-2,7	2,76061
C4,5 - Markov11	*	8,67444	3,13748
C4,5 - Markov15	*	9,90244	3,13748
C4,5 - Markov25	*	8,80344	3,13748
C4,5 - RandomForest		-2,686	2,76061
C4,5 - SmoPoly		1,79467	2,76061
C4,5 - TAN		0,477333	2,76061
Furia - KNN-1	*	-3,82667	2,76061
Furia - Markov11	*	7,54778	3,13748
Furia - Markov15	*	8,77578	3,13748
Furia - Markov25	*	7,67678	3,13748
Furia - RandomForest	*	-3,81267	2,76061
Furia - SmoPoly		0,668	2,76061
Furia - TAN		-0,649333	2,76061
KNN-1 - Markov11	*	11,3744	3,13748
KNN-1 - Markov15	*	12,6024	3,13748
KNN-1 - Markov25	*	11,5034	3,13748
KNN-1 - RandomForest		0,014	2,76061
KNN-1 - SmoPoly	*	4,49467	2,76061
KNN-1 - TAN	*	3,17733	2,76061
Markov11 - Markov15		1,228	3,38105
Markov11 - Markov25		0,129	3,38105
Markov11 - RandomForest	*	-11,3604	3,13748
Markov11 - SmoPoly	*	-6,87978	3,13748
Markov11 - TAN	*	-8,19711	3,13748
Markov15 - Markov25		-1,099	3,38105
Markov15 - RandomForest	*	-12,5884	3,13748
Markov15 - SmoPoly	*	-8,10778	3,13748
Markov15 - TAN	*	-9,42511	3,13748
Markov25 - RandomForest	*	-11,4894	3,13748
Markov25 - SmoPoly	*	-7,00878	3,13748

Markov25 - TAN	*	-8,32611	3,13748
RandomForest - SmoPoly	*	4,48067	2,76061
RandomForest - TAN	*	3,16333	2,76061
SmoPoly - TAN		-1,31733	2,76061

\* indica una diferencia significativa.

Esta tabla aplica un procedimiento de comparación múltiple para determinar cuáles medias son significativamente diferentes de otras. La mitad inferior de la salida muestra las diferencias estimadas entre cada par de medias. El asterisco que se encuentra al lado de los 24 pares indica que estos pares muestran diferencias estadísticamente significativas con un nivel del 95,0% de confianza.

Podemos ver esta información de forma gráfica en la siguiente figura, donde se indican cada uno de los métodos en el eje X:



## 6.2.2 Análisis estadístico del tiempo de computación con dos categorías en la variable de salida

En esta ocasión, el tiempo de computación del sistema en la clasificación de intrusos va a ser la variable estudiada, que vamos a denominar como variable dependiente: *TiemTr*, siendo nuestros tres factores o efectos principales:

1. Tipo de Filtro (Selección de Atributos )
2. Tipo de Discretización (Discr)

### 3. Tipo de Algoritmo (ALG)

En primer lugar, se va a realizar el estudio de la tabla ANOVA. Este procedimiento ejecuta un análisis de varianza de varios factores para TiemTr-

#### Análisis de Varianza para TiemTr - Suma de Cuadrados Tipo III

<i>Fuente</i>	<i>Suma de Cuadrados</i>	<i>Gl</i>	<i>Cuadrado Medio</i>	<i>Razón-F</i>	<i>Valor-P</i>
EFFECTOS PRINCIPALES					
A:Filtro	543,456	4	135,864	5,89	0,0003
B:Discr	341,417	2	170,709	7,40	0,0010
C:ALG	4707,52	8	588,44	25,52	0,0000
RESIDUOS	2421,21	105	23,0591		
TOTAL (CORREGIDO)	8571,13	119			

Todas las razones-F se basan en el cuadrado medio del error residual

La tabla ANOVA descompone la variabilidad de TiemTr en contribuciones debidas a varios factores. Puesto que se ha escogido la suma de cuadrados Tipo III (por omisión), la contribución de cada factor se mide eliminando los efectos de los demás factores. Los valores-P prueban la significancia estadística de cada uno de los factores. Puesto que 3 valores-P son menores que 0,05, estos factores tienen un efecto estadísticamente significativo sobre el tiempo de computación (TiemTr) con un 95,0% de nivel de confianza.

Por tanto, como primera conclusión, se puede decir que los tres efectos principales tienen una repercusión relevante sobre la precisión global del sistema de clasificación de detección de intrusos. Aunque los tres valores de Valor-P son menores que 0.05, lo que ahora es interesante de analizar, es si existen diferentes valores de Filtro, o de Discretización o de Algoritmo, que se comporten de forma similar, y por lo tanto, intentar obtener grupos homogéneos o heterogéneos de estos tres efectos o factores. Como conclusión global los tres factores (los niveles que se seleccionen para los mismos) influyen significativamente en el tiempo de computación. Para realizar un análisis más detallado, vamos a proceder a realizar la prueba de múltiples rangos.

Recordemos, como se explicó para el caso de AcGlobal que con la realización de estas tablas de múltiples rangos, se realizan comparaciones múltiples para determinar cuáles medias son significativamente diferentes de otras. Las que sean homogéneas o similares desde el punto de vista estadístico, tienen una X en la misma columna, y por

tanto se pueden considerar como un grupo homogéneo. Si existen X en diversas columnas, significa que existen más de un grupo. Esto conlleva a que por tanto este factor sea estadísticamente significativo, al igual que se ha realizado en la sección anterior.

### 6.2.2.1 Estudio de tablas de múltiple rangos para tiempo computacional

Comenzamos estudiando el factor Filtro, siendo la tabla de rangos múltiples:

Método: 95,0 porcentaje LSD

<i>Filtro</i>	<i>Casos</i>	<i>Media LS</i>	<i>Sigma LS</i>	<i>Grupos Homogéneos</i>
fcfs	24	3,37111	0,991034	X
fnb	24	3,70986	0,991034	XX
fcns	24	4,34236	0,991034	XX
fc45	24	6,22028	0,991034	X
All	24	9,12694	0,991034	X

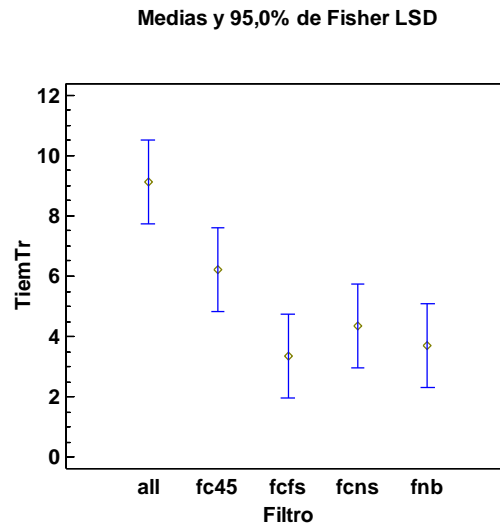
<i>Contraste</i>	<i>Sig.</i>	<i>Diferencia</i>	<i>+/- Límites</i>
all - fc45	*	2,90667	2,74862
all - fcfs	*	5,75583	2,74862
all - fcns	*	4,78458	2,74862
all - fnb	*	5,41708	2,74862
fc45 - fcfs	*	2,84917	2,74862
fc45 - fcns		1,87792	2,74862
fc45 - fnb		2,51042	2,74862
fcfs - fcns		-0,97125	2,74862
fcfs - fnb		-0,33875	2,74862
fcns - fnb		0,6325	2,74862

\* indica una diferencia significativa.

Como se puede apreciar, existen tres grupos de filtros desde el punto de vista estadístico, con intersección entre los mismos. El primer grupo lo constituyen los filtros denominados: FCFS, FNB y FCNS. El segundo grupo lo constituyen los filtros: FNB, FCNS y FC4.5. Y finalmente, el último grupo lo compone el filtro denominado: ALL.

Se puede ver como el primer grupo, el compuesto por FCFS, FNB y FCNS, es el que estadísticamente tiene en media un menor tiempo de computación en las pruebas experimentales realizadas. Por el contrario, el grupo tercero, compuesto por el filtro con

nombre ALL- recordemos que en este caso no se aplica selección de atributos-, es el que requiere un mayor tiempo de computación. Como conclusión la selección de atributos en mejora bastante el tiempo de construcción del modelo. De forma gráfica obtenemos la siguiente información sobre las medias y desviaciones del tipo de filtro:



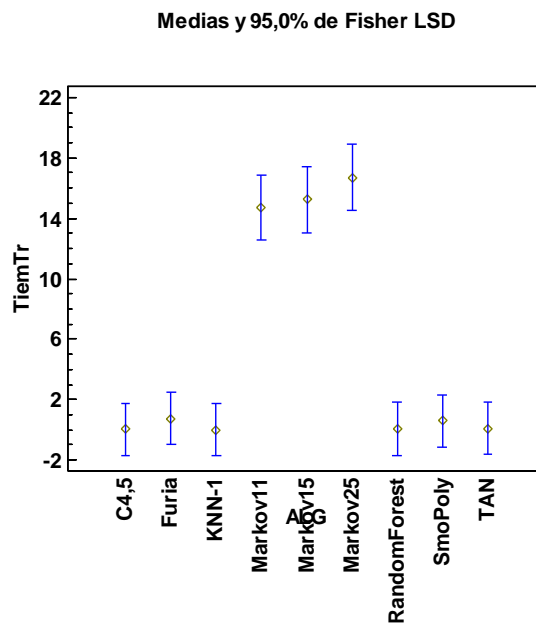
Se realiza ahora el test de rangos múltiples para el factor tipo de algoritmo, obteniendo los resultados:

**Método: 95,0 porcentaje LSD**

<i>ALG</i>	<i>Casos</i>	<i>Media LS</i>	<i>Sigma LS</i>	<i>Grupos Homogéneos</i>
KNN-1	15	0	1,23987	×
C4,5	15	0,042	1,23987	×
RandomForest	15	0,0686667	1,23987	×
TAN	15	0,0933333	1,23987	×
SmoPoly	15	0,586	1,23987	×
Furia	15	0,746	1,23987	×
Markov11	10	14,7087	1,56013	×
Markov15	10	15,2387	1,56013	×
Markov25	10	16,7037	1,56013	×

Se pueden apreciar dos grupos homogéneos sin intersección entre ambos. El primer grupo lo componen los algoritmos: KNN-1, C4.5, RANDOM FOREST, TAN,

SmoPoly y FURIA. El segundo grupo lo componen los algoritmos: MARKOV11, MARKOV15 y MARKOV25. Indicar, que este último grupo es el que requiere mayor tiempo de computación, puesto que para los modelos de Markov se debe construir un modelo para cada categoría. De forma gráfica, la siguiente figura muestra esta información:



Finalmente, para el factor tipo de discretización, los resultados obtenidos para la tabla de rangos múltiples se presentan a continuación:

Método: 95,0 porcentaje LSD

<i>Discr</i>	<i>Casos</i>	<i>Media LS</i>	<i>Sigma LS</i>	<i>Grupos Homogéneos</i>
fay	45	3,36756	0,715839	x
0	30	5,43344	0,946966	xx
fre	45	7,26133	0,715839	x

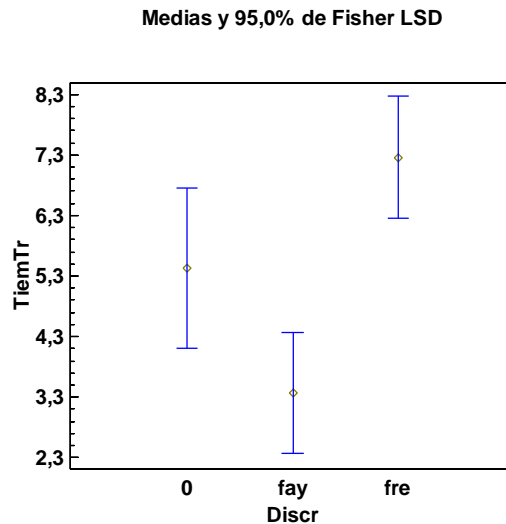
<i>Contraste</i>	<i>Sig.</i>	<i>Diferencia</i>	<i>+/- Límites</i>
0 – fay		2,06589	2,35377
0 – fre		-1,82789	2,35377
fay – fre	*	<b>-3,89378</b>	2,00731

\* indica una diferencia significativa.

Existen dos grupos homogéneos, con intersección entre los mismos. El primer grupo está formado por los tipos de discretización: FAY –Fayyad & Irani- y 0 -no se aplica discretización-. El segundo grupo lo forma: 0 y FRE –intervalos de igual frecuencia-. El



primer grupo es el que estadísticamente tiene un menor tiempo de computación, siendo la discretización FAY la que tiene menor valor en esta variable, con una media de 3.36. Por el contrario, el método FRE es el que requiere mayor tiempo, con un valor de 7.26.



### 6.3 Experimentos donde existen cinco categorías en la variable de clasificación.

Para este apartado se seguirá la misma metodología, que la utilizada en la sección anterior. No obstante, hay que destacar que en este caso la variable de salida para la clasificación es más específica, puesto que los ataques son clasificados en cinco clases diferentes. Además, se han considerado un número mayor de variantes para los tipos de algoritmos.

#### 6.3.1 Análisis estadístico del error global con cinco categorías en la variable de salida

Vamos a comenzar analizando el error global del sistema en la clasificación de intrusos, que vamos a denominar como variable dependiente: AcGlobal, siendo nuestros tres factores o efectos principales:

1. Tipo de Selección de atributos (Filtro)
2. Tipo de Discretización (Discr)

### 3. Tipo de Algoritmo (ALG)

Vamos a comenzar como en la sección anterior, realizando el estudio ANOVA. Este procedimiento ejecuta un análisis de varianza de varios factores para AcGlobal. Realiza varias pruebas y gráficas para determinar qué factores tienen un efecto estadísticamente significativo sobre AcGlobal. También evalúa la significancia de las interacciones entre los factores, si es que hay suficientes datos. Las pruebas-F en la tabla ANOVA le permitirán identificar los factores significativos.

#### Análisis de Varianza para AcGlobal - Suma de Cuadrados Tipo III

<i>Fuente</i>	<i>Suma de Cuadrados</i>	<i>Gl</i>	<i>Cuadrado Medio</i>	<i>Razón-F</i>	<i>Valor-P</i>
EFFECTOS PRINCIPALES					
A:Filtro	827,943	4	206,986	2,46	0,0458
B:Discr	5438,28	2	2719,14	32,29	0,0000
C:ALG	41924,9	21	1996,42	23,71	0,0000
RESIDUOS	24166,6	287	84,2041		
TOTAL (CORREGIDO)	71471,6	314			

Todas las razones-F se basan en el cuadrado medio del error residual

La tabla ANOVA descompone la variabilidad de AcGlobal en contribuciones debidas a varios factores. Puesto que se ha escogido la suma de cuadrados Tipo III (por omisión), la contribución de cada factor se mide eliminando los efectos de los demás factores. Los valores-P prueban la significancia estadística de cada uno de los factores. Puesto que 3 valores-P son menores que 0,05, estos factores tienen un efecto estadísticamente significativo sobre AcGlobal con un 95,0% de nivel de confianza. Por tanto, todos los factores son significativos y ahora es necesario analizar la tabla de rangos múltiples para cada uno de los efectos principales.

#### 6.3.1.1 Estudio de tablas de múltiple rangos para ACGlobal

Comenzamos estudiando el factor Algoritmo, siendo la tabla de rangos múltiples la siguiente:

Método: 95,0 porcentaje LSD

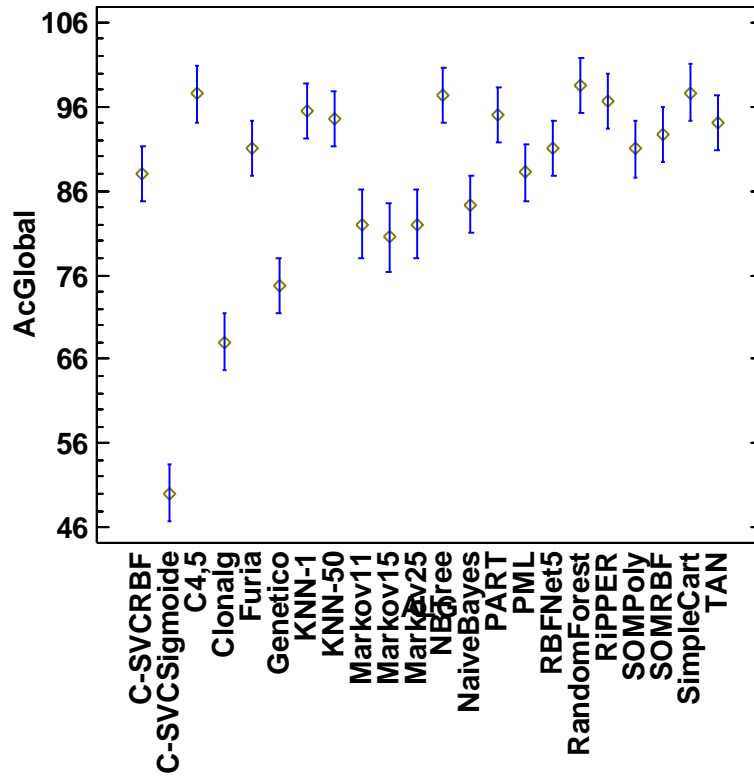
<i>ALG</i>	<i>Casos</i>	<i>Media LS</i>	<i>Sigma LS</i>	<i>Grupos Homogéneos</i>
C-SVCSigmoide	15	50,098	2,3693	X
Clonalg	15	68,0393	2,3693	X
Genetico	15	74,6613	2,3693	X
Markov15	10	80,5217	2,92714	XX
Markov11	10	81,9587	2,92714	XXX
Markov25	10	81,9747	2,92714	XXX
NaiveBayes	15	84,396	2,3693	XXX
C-SVCRBF	15	87,9347	2,3693	XXX
PML	15	88,1327	2,3693	XXX
SOMPoly	15	90,9393	2,3693	XXX
RBFNet5	15	91,1013	2,3693	XXX
Furia	15	91,116	2,3693	XXX
SOMRBF	15	92,6387	2,3693	XXXX
TAN	15	94,0107	2,3693	XXXX
KNN-50	15	94,504	2,3693	XXXX
PART	15	95,0313	2,3693	XXX
KNN-1	15	95,468	2,3693	XXX
RiPPER	15	96,656	2,3693	XXX
NBTree	15	97,2973	2,3693	XXX
C4.5	15	97,466	2,3693	XXX
SimpleCart	15	97,6873	2,3693	XX
RandomForest	15	98,4707	2,3693	X

Como se puede ver en esta tabla, existes diez grupos diferentes en los que se pueden agrupar los algoritmos utilizados, mencionando que no existen diferencias estadísticamente significativas entre aquellos niveles que compartan una misma columna de X's. Como algunos métodos pertenecen a más de un grupo, existe por tanto intersección entre los mismos.

En el primer grupo comentar que estaría el método: C-SVCSigmoide –SVM con kernel sigmoideo-. En el grupo segundo el método: Clonalg –Algoritmo Inmune Artificial-. En el tercer grupo las diferentes variantes de Modelos de Markov: MARKOV15, MaARKOV25 y MARKOV11, junto con el genético. De forma sucesiva se pueden mencionar los restantes grupos, hasta llegar al último que tendría los algoritmos: SOMRBF, TAN, KNN-50, PART, KNN-1, RIPPER, NBTREE, C4.5, CART, RANDOM FOREST, siendo este grupo el que mejor resultados obtiene. Como

conclusión importante, indicar que este grupo es el que mejor resultados obtiene, siendo el RANDOM FOREST el mejor algoritmo.

**Medias y 95,0% de Fisher LSD**



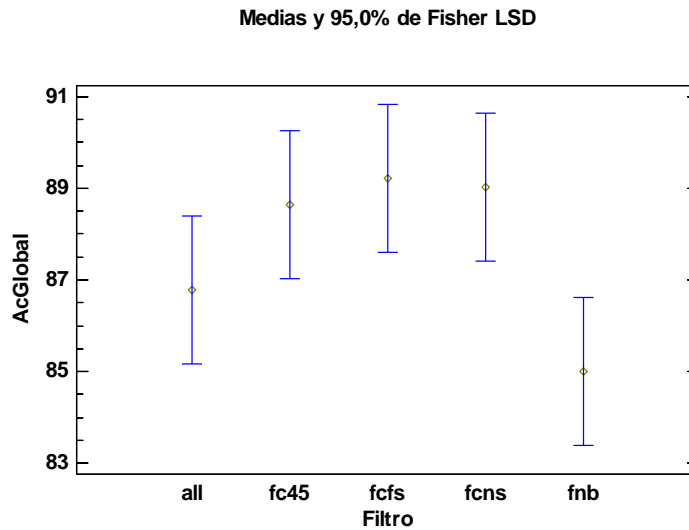
Continuamos el análisis con tipo de Filtro para la variable precisión, obteniendo la siguiente tabla de rangos múltiples:

**Método: 95,0 porcentaje LSD**

Filtro	Casos	Media LS	Sigma LS	Grupos Homogéneos
Fnb	63	84,9926	1,15955	X
All	63	86,7761	1,15955	XX
fc45	63	88,6405	1,15955	X
Fcns	63	89,0375	1,15955	X
Fcfs	63	89,2132	1,15955	X

Por tanto hay dos grupos: un primer grupo que tendría al tipo de filtro FNB y ALL, que sería el que peor valor de precisión obtendría, ya que la media de FNB es de 84,9 % de clasificación, y un segundo grupo homogéneo, donde tendría los restantes tipos de

filtros: ALL, FC45, FCNS, FCFS. Es decir, desde el punto de vista estadístico, y para la variable de salida de precisión global, estos cuatro tipos de filtros son equivalentes o de forma similar, tienen el mismo comportamiento, y el que tiene mejor resultado de clasificación es FCFS. Estos resultados de forma gráfica se presentan a continuación:



Para la variable tipo de discretización, se obtienen los siguientes resultados:

Método: 95,0 porcentaje LSD

<i>Discr</i>	<i>Casos</i>	<i>Media LS</i>	<i>Sigma LS</i>	<i>Grupos Homogéneos</i>
0	95	81,5833	0,973033	x
Fre	110	90,3253	0,874924	x
Fay	110	91,2874	0,874924	x

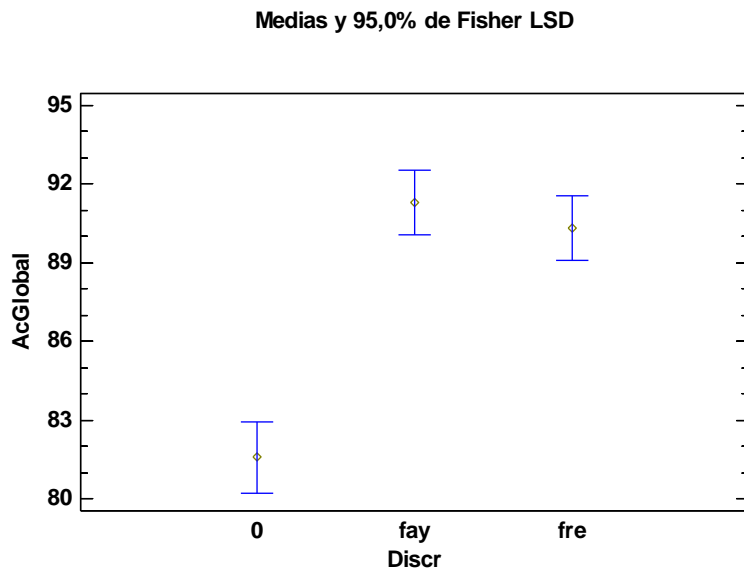
<i>Contraste</i>	<i>Sig.</i>	<i>Diferencia</i>	<i>+/- Límites</i>
0 - fay	*	-9,70405	2,57556
0 - fre	*	-8,74195	2,57556
fay - fre		0,962091	2,4354

\* indica una diferencia significativa.

Como se ha indicado, esta tabla aplica un procedimiento de comparación múltiple para determinar cuáles medias son significativamente diferentes de otras. La mitad inferior de la salida muestra las diferencias estimadas entre cada par de medias.

El asterisco que se encuentra al lado de los 2 pares indica que estos pares muestran diferencias estadísticamente significativas con un nivel del 95,0% de confianza. En la parte superior de la página, se han identificado 2 grupos homogéneos según la alineación de las X's en columnas. No existen diferencias estadísticamente significativas entre aquellos niveles que compartan una misma columna de X's. El método empleado actualmente para discriminar entre las medias es el procedimiento de diferencia mínima significativa (LSD) de Fisher. Con este método hay un riesgo del 5,0% al decir que cada par de medias es significativamente diferente, cuando la diferencia real es igual a 0.

Observando la tabla de rangos se ver claramente la existencia de dos grupos: el primero lo forma el tipo de discretización 0 –ninguna discretización aplicada-, y el segundo grupo lo forman el tipo de discretización: FRE -igual intervalo de frecuencias- y FAY –Fayyad & Irani-. Este grupo es el que mejores prestaciones obtiene, siendo FAY, discretización supervisada basada en entropía, el mejor tipo de discretización.



### 6.3.2 Análisis estadístico del tiempo de computación con cinco categorías en la variable de salida

El tiempo de computación del sistema en la clasificación de intrusos va a ser la variable estudiada, que vamos a denominar como variable dependiente: *TiemTr*, siendo nuestros tres factores o efectos principales:

1. Tipo de Selección de atributos (Filtro)
2. Tipo de Discretización (Discr)
3. Tipo de Algoritmo (ALG)

Como se viene haciendo en primer lugar, se va a realizar el estudio de la tabla ANOVA. Este procedimiento ejecuta un análisis de varianza de varios factores para *TiemTr*. Indicar que en este experimento, el número de casos completos o diferentes combinaciones estudiadas fue de 315 casos.

#### Análisis de Varianza para *TimeTr* - Suma de Cuadrados Tipo III

<i>Fuente</i>	<i>Suma de Cuadrados</i>	<i>Gl</i>	<i>Cuadrado Medio</i>	<i>Razón-F</i>	<i>Valor-P</i>
EFFECTOS PRINCIPALES					
A:Filtro	2,23405E8	4	5,58512E7	1,33	0,2586
B:Discr	1,92204E8	2	9,61022E7	2,29	0,1031
C:ALG	5,59739E9	21	2,66543E8	6,35	0,0000
RESIDUOS	1,20455E10	287	4,19703E7		
TOTAL (CORREGIDO)	1,80374E10	314			

Todas las razones-F se basan en el cuadrado medio del error residual

Los resultados de esta tabla ANOVA indican claramente como el tipo de Filtro y el tipo de Discretización tienen un efecto muy pequeño y por tanto, no estadísticamente significativo sobre el tiempo requerido de computación. Sin embargo, el tipo de Algoritmo si es relevante, teniendo un valor de VALOR-P inferior a 0.05.

#### 6.3.2.1 Estudio de tablas de múltiple rangos para tiempo computacional

Comenzamos estudiando el factor Algoritmo, siendo la tabla de rangos múltiples

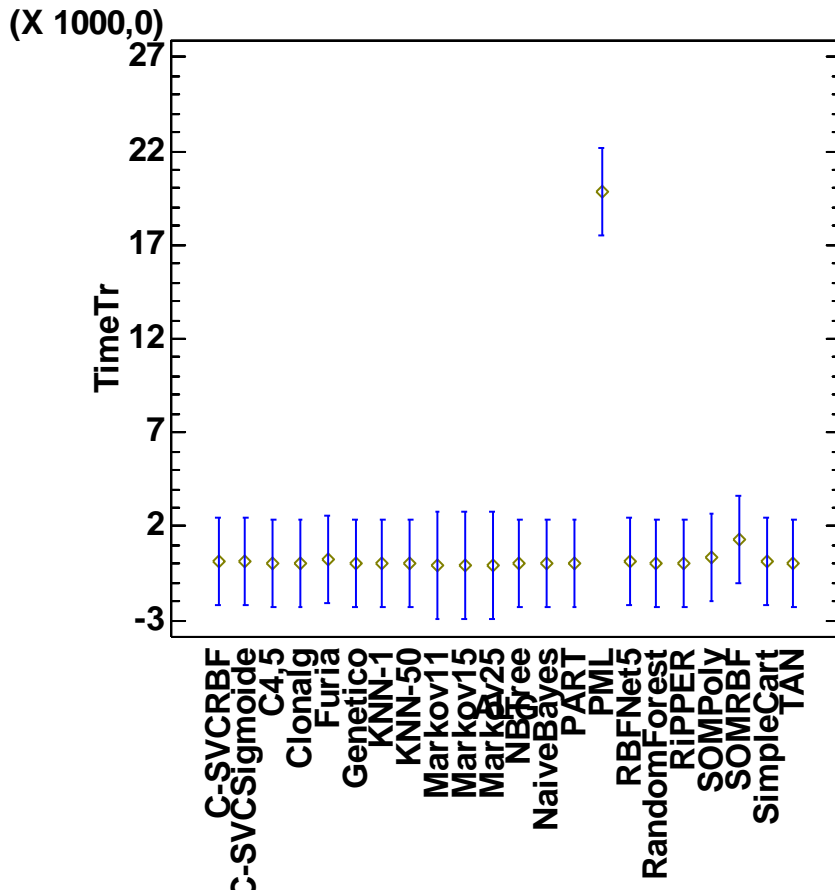
Método: 95,0 porcentaje LSD

<i>ALG</i>	<i>Casos</i>	<i>Media LS</i>	<i>Sigma LS</i>	<i>Grupos Homogéneos</i>
Markov11	10	-78,6079	2066,56	X
Markov15	10	-73,1929	2066,56	X
Markov25	10	-62,6419	2066,56	X
KNN-1	15	0	1672,73	X
KNN-50	15	0	1672,73	X
NaiveBayes	15	0,139333	1672,73	X
RandomForest	15	1,28076	1672,73	X
C4.5	15	1,424	1672,73	X
TAN	15	1,56267	1672,73	X
PART	15	4,37667	1672,73	X
Clonalg	15	6,93467	1672,73	X
NBTree	15	36,1947	1672,73	X
RiPPER	15	37,6733	1672,73	X
Genetico	15	60,98	1672,73	X
SimpleCart	15	104,928	1672,73	X
C-SVCSigmoide	15	143,942	1672,73	X
RBFNet5	15	147,378	1672,73	X
C-SVCRBF	15	161,329	1672,73	X
Furia	15	265,101	1672,73	X
SOMPoly	15	316,344	1672,73	X
SOMRBF	15	1272,4	1672,73	X
PML	15	19876,5	1672,73	X

Como se puede ver, existen dos grupos homogéneos. Un primer grupo donde están todos los algoritmos analizados, salvo el algoritmo PML-PERCEPTRÓN MULTICAPA-. Un segundo grupo donde está por tanto PML, que es un algoritmo que requiere mucho más tiempo de computación que los restantes. Si se visualiza de forma gráfica esta información, se puede ver claramente que PML es muy superior en tiempo de computación:



### Medias y 95,0% de Fisher LSD



A continuación se presentan las pruebas de Múltiple Rangos para TimeTr para la variable Filtro:

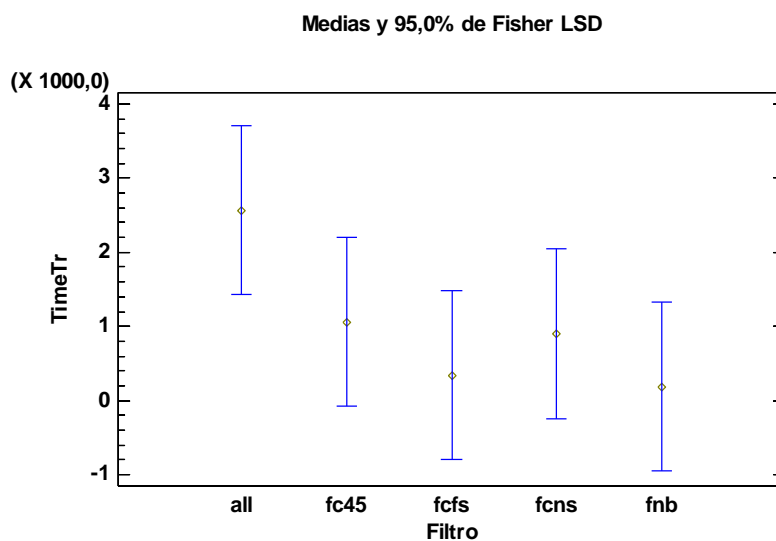
Método: 95,0 porcentaje LSD

Filtro	Casos	Media LS	Sigma LS	Grupos Homogéneos
fnb	63	188,934	818,645	X
fcfs	63	341,891	818,645	XX
fcns	63	900,384	818,645	XX
fc45	63	1056,84	818,645	XX
all	63	2562,86	818,645	X

Contraste	Sig.	Diferencia	+/- Límites
all - fc45		1506,02	2271,96
all - fcfs		2220,97	2271,96
all - fcns		1662,48	2271,96
all - fnb	*	2373,93	2271,96
fc45 - fcfs		714,948	2271,96
fc45 - fcns		156,456	2271,96
fc45 - fnb		867,906	2271,96
fcfs - fcns		-558,492	2271,96
fcfs - fnb		152,957	2271,96
fcns - fnb		711,45	2271,96

\* indica una diferencia significativa.

Existen dos grupos con intersección. El primer grupo está constituido por FNB, FCFS, FCNS, FC4.5. Este primer grupo es el que requiere menor tiempo de computación. El segundo grupo está formado por: FCFS, FCNS, FC4.5 y ALL. La selección de atributos de tipo wrapper FNB es la que menor tiempo consume.



Finalmente, se realiza las pruebas de Múltiple Rangos para TimeTr para la variable Discr.

Método: 95,0 porcentaje LSD

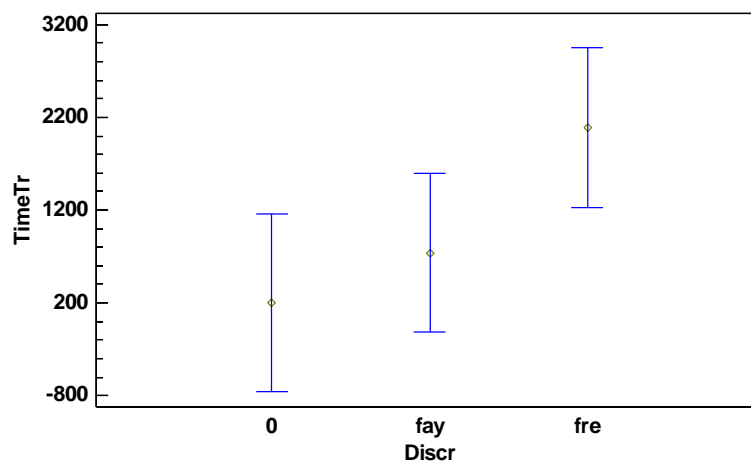
<i>Discr</i>	<i>Casos</i>	<i>Media LS</i>	<i>Sigma LS</i>	<i>Grupos Homogéneos</i>
0	95	203,356	686,961	x
Fay	110	741,281	617,696	xx
Fre	110	2085,91	617,696	x

<i>Contraste</i>	<i>Sig.</i>	<i>Diferencia</i>	<i>+/- Límites</i>
0 - fay		-537,924	1818,35
0 - fre	*	<b>-1882,55</b>	1818,35
fay - fre		-1344,63	1719,39

\* indica una diferencia significativa.

En este caso cabe decir que la discretización no ha mejorado mucho el tiempo de construcción de los modelos. Pero comparando la discretización de tipo Fayyad –FAY- y la de igual intervalos de Frecuencia sí cabe mencionar que existe una diferencia bastante considerable siendo la discretización de tipo Fayyad la que menor tiempo consume.

Medias y 95,0% de Fisher LSD



## 6.4 Experimentos donde existen veinte categorías en la variable de saluda de clasificación. Estudio a nivel de ataque.

En este apartado, hay que destacar que en este caso la variable de salida para la clasificación es más específica, puesto que los ataques son clasificados en 20 clases diferentes. Además, se han considerado el mismo número de variantes para los tipos de algoritmos que el utilizado en el primer caso de estudio a nivel de 2 categorías.

### 6.4.1 Análisis Estadístico del error global con 20 categorías en la variable de salida

Vamos a comenzar analizando el error global del sistema en la clasificación de intrusos, que vamos a denominar como variable dependiente: AcGlobal, siendo nuestros tres factores o efectos principales:

- Tipo de Selección de atributos (Filtro)
- Tipo de Discretización (Discr)
- Tipo de Algoritmo (ALG)

Como en secciones anteriores se ha realizado el estudio ANOVA para el análisis de varianza de varios factores para AcGlobal.

#### Análisis de Varianza para AcGlobal - Suma de Cuadrados Tipo III

<i>Fuente</i>	<i>Suma de Cuadrados</i>	<i>Gl</i>	<i>Cuadrado Medio</i>	<i>Razón-F</i>	<i>Valor-P</i>
EFFECTOS PRINCIPALES					
A:Filtro	5762,03	4	1440,51	144,66	0,0000
B:Discr	70,6078	2	35,3039	3,55	0,0324
C:ALG	782,157	8	97,7696	9,82	0,0000
RESIDUOS	1045,6	105	9,95809		
TOTAL (CORREGIDO)	7697,12	119			

Todas las razones-F se basan en el cuadrado medio del error residual

Los valores-P prueban la significancia estadística de cada uno de los factores. Puesto que 3 valores-P son menores que 0,05, estos factores tienen un efecto estadísticamente significativo sobre AcGlobal con un 95,0% de nivel de confianza.

Como en casos anteriores, aquí no podemos apreciar si hay diferentes grupo homogéneos o heterogéneos con intersección vacía o no entre los diferentes niveles. Para ello se hace un análisis de tablas de rangos múltiples.

#### 6.4.1.1 Estudio de tablas de múltiple rangos para ACGlobal

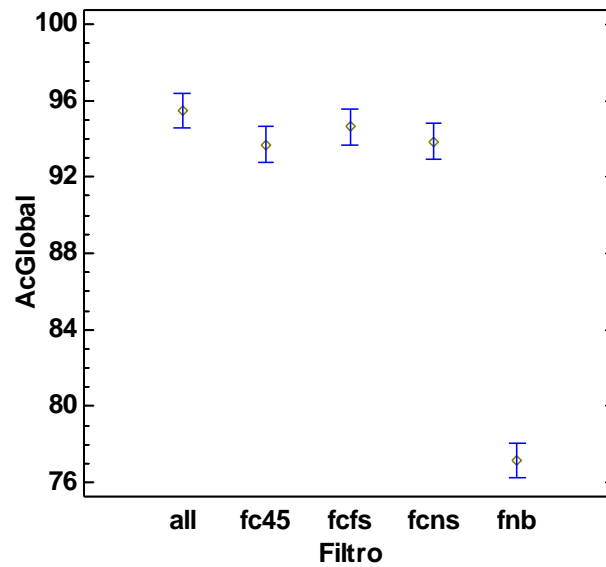
Comenzamos estudiando el factor FILTRO, siendo la tabla de rangos múltiples la siguiente:

Método: 95,0 porcentaje LSD

<i>Filtro</i>	<i>Casos</i>	<i>Media LS</i>	<i>Sigma LS</i>	<i>Grupos Homogéneos</i>
fnb	24	77,1701	0,651261	x
fc45	24	93,7005	0,651261	x
fcns	24	93,8689	0,651261	x
fcfs	24	94,6139	0,651261	x
all	24	95,4989	0,651261	x

Existen dos grupos, el primero es FNB, y el Segundo contiene todas las restantes formas de filtro: FC45, FCNS, FCFS y ALL. El segundo grupo es el que mejores resultados obtiene, siendo ALL en concreto el que tiene un valor en media de clasificación superior al resto, con el 95.49%.

### Medias y 95,0% de Fisher LSD



Continuamos con el estudio de tipo de Discretización:

### Método: 95,0 porcentaje LSD

Discr	Casos	Media LS	Sigma LS	Grupos Homogéneos
fay	45	90,1764	0,470416	X
0	30	90,8029	0,622301	XX
fre	45	91,932	0,470416	X

Contraste	Sig.	Diferencia	+/- Límites
0 - fay		0,626444	1,54679
0 - fre		-1,12911	1,54679
fay - fre	*	<b>-1,75556</b>	1,31911

\* indica una diferencia significativa.

Para discretización se obtienen dos grupos: el primero formado por FAY –método de Fayyad & Irani- y 0 -no se ha aplicado discretización-. El segundo grupo formado por 0 y FRE –método no supervisado de intervalos de igual frecuencia-. El segundo grupo obtiene los mejores resultados de clasificación, y concretamente FRE obtiene un 91.932%.

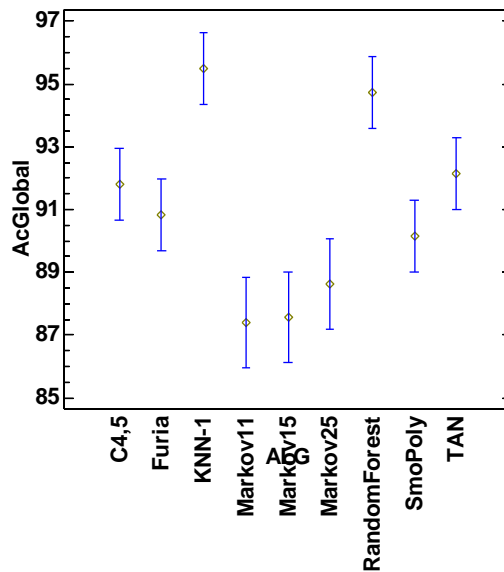
Finalmente se realiza el estudio para tipo de Algoritmo, obteniendo los siguientes resultados:

Método: 95,0 porcentaje LSD

ALG	Casos	Media LS	Sigma LS	Grupos Homogéneos
Markov11	10	87,3862	1,02525	x
Markov15	10	87,5752	1,02525	xx
Markov25	10	88,6452	1,02525	xxx
SmoPoly	15	90,16	0,814784	xxx
Furia	15	90,8333	0,814784	xx
C4,5	15	91,7927	0,814784	x
TAN	15	92,1447	0,814784	x
RandomForest	15	94,7173	0,814784	x
KNN-1	15	95,4793	0,814784	x

Existen cinco grupos, siendo el primero compuesto por las diferentes variantes de MARKOV: MARKOV11, MARKOV15 y MARKOV25. Este grupo primero es el que peores prestaciones presenta en cuanto al porcentaje global de clasificación obtenido. El segundo grupo MARKOV15, MARKOV25 y SMOPOLY, y así sucesivamente hasta el último grupo, que lo forma: RANDOM FOREST y KNN-1. Este grupo es el que tiene mejores prestaciones en cuanto al error, y en concreto, KNN-1 obtiene unos resultados del orden del 95.47%.

Medias y 95,0% de Fisher LSD



## 6.4.2 Análisis estadístico del tiempo de computación con 20 categorías en la variable de salida.

Vamos a comenzar analizando el tiempo de computación del sistema en la clasificación de intrusos, que vamos a denominar como variable dependiente: TimeTr, siendo nuestros tres factores o efectos principales:

- Tipo de selección de características (Filtro)
- Tipo de Discretización (Discr)
- Tipo de Algoritmo (ALG)

La siguiente tabla muestra el estudio ANOVA para TimeTr:

### Análisis de Varianza para TimeTr - Suma de Cuadrados Tipo III

Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P
EFECTOS PRINCIPALES					
A:Filtro	578,028	4	144,507	5,28	0,0007
B:Discr	572,148	2	286,074	10,45	0,0001
C:ALG	5515,66	8	689,457	25,18	0,0000
RESIDUOS	2874,8	105	27,379		
TOTAL (CORREGIDO)	10201,9	119			

Todas las razones-F se basan en el cuadrado medio del error residual

### 6.4.2.1 Estudio de tablas de múltiple rangos para tiempo computacional

#### Pruebas de Múltiple Rangos para TimeTr por ALG

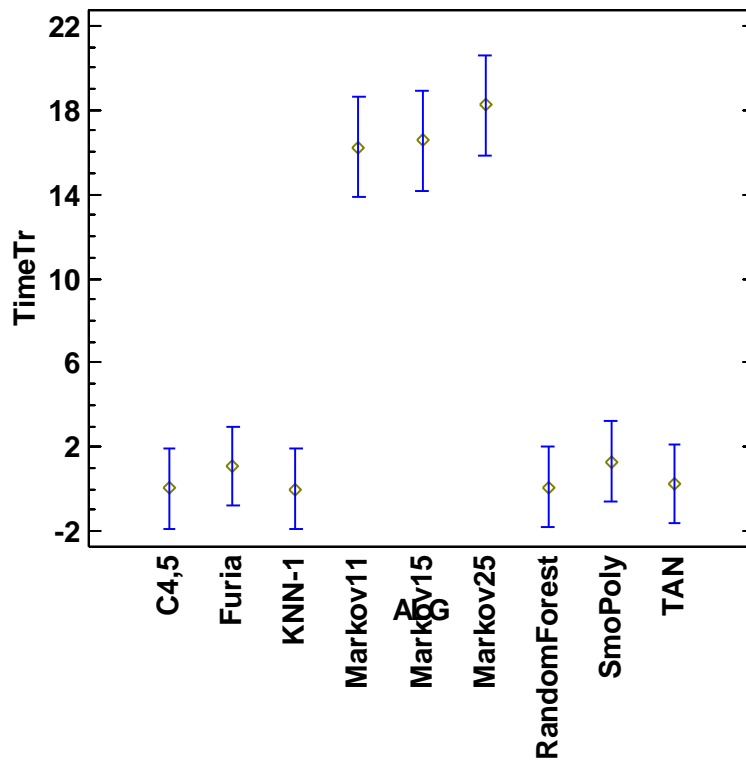
Método: 95,0 porcentaje LSD

ALG	Casos	Media LS	Sigma LS	Grupos Homogéneos
KNN-1	15	0	1,35102	x
C4,5	15	0,034	1,35102	x
RandomForest	15	0,107333	1,35102	x
TAN	15	0,250667	1,35102	x
Furia	15	1,08133	1,35102	x
SmoPoly	15	1,30467	1,35102	x
Markov11	10	16,2097	1,7	x
Markov15	10	16,5327	1,7	x
Markov25	10	18,2457	1,7	x



En la tabla se puede observar la existencia de dos grupos claramente diferenciados, se puede deducir que el algoritmo que menos tiempo consume es el KNN-1, seguido de árboles de decisión C4.5 y Random Forest y el que más consume son los modelos ocultos de Markov, más precisamente el que utiliza 25 estados. Gráficamente:

**Medias y 95,0% de Fisher LSD**



Pruebas de Múltiple Rangos para TimeTr por Filtro

Método: 95,0 porcentaje LSD

Filtro	Casos	Media LS	Sigma LS	Grupos Homogéneos
fcfs	24	4,02017	1,07988	X
fnb	24	4,1035	1,07988	X
fcns	24	5,02142	1,07988	X
fc45	24	6,87058	1,07988	XX
all	24	9,85433	1,07988	X

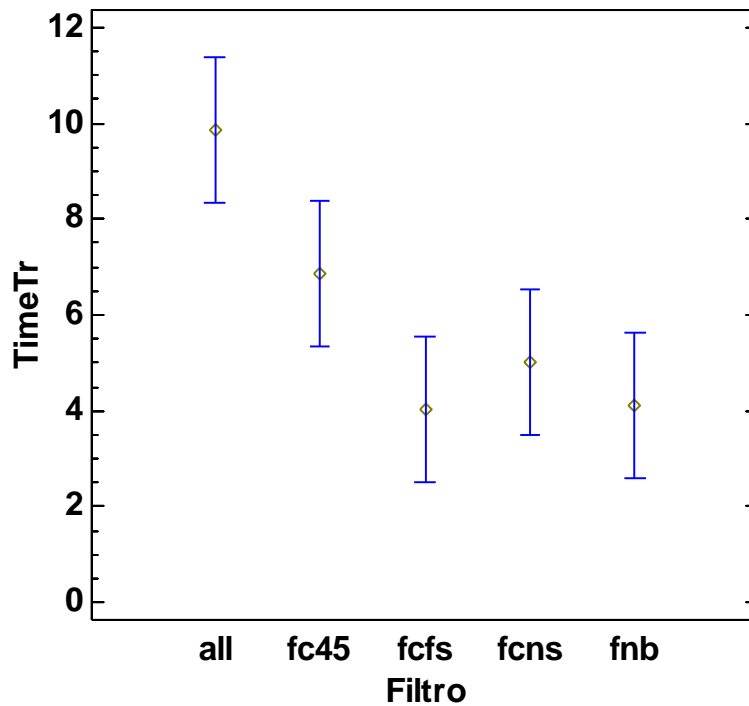
Contraste	Sig.	Diferencia	+/- Límites
all - fc45		2,98375	2,99503
all - fcfs	*	5,83417	2,99503
all - fcns	*	4,83292	2,99503

all – fnb	*	5,75083	2,99503
fc45 – fcfs		2,85042	2,99503
fc45 - fcns		1,84917	2,99503
fc45 – fnb		2,76708	2,99503
fcfs – fcns		-1,00125	2,99503
fcfs – fnb		-0,0833333	2,99503
fcns – fnb		0,917917	2,99503

\* indica una diferencia significativa.

Se deduce que la selección de atributos influye bastante en el tiempo de construcción del modelo siendo la del tipo de filtro fcfs la que menor tiempo requiero y sin selección de atributos “all” el clasificador requiere de más tiempo. Gráficamente:

### Medias y 95,0% de Fisher LSD



## Pruebas de Múltiple Rangos para TimeTr por Discr

Método: 95,0 porcentaje LSD

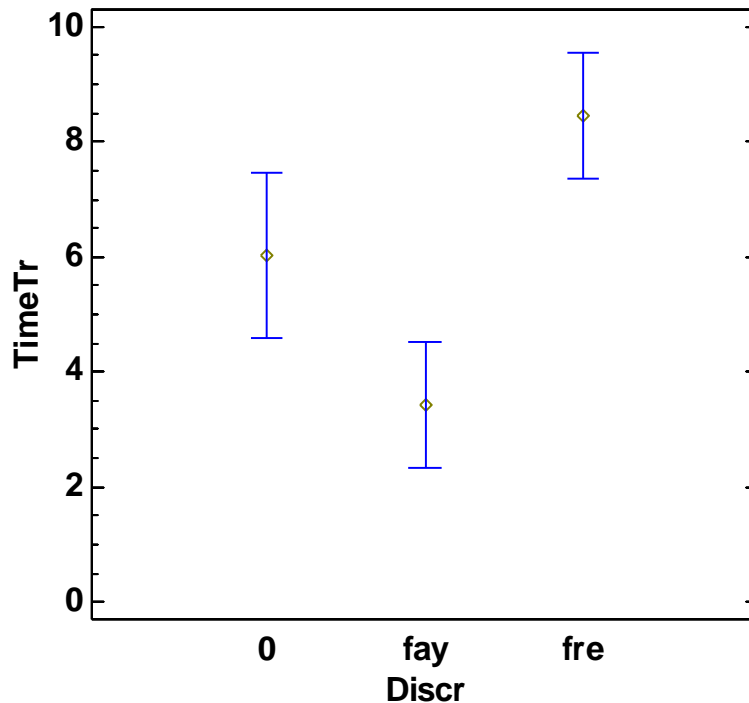
<i>Discr</i>	<i>Casos</i>	<i>Media LS</i>	<i>Sigma LS</i>	<i>Grupos Homogéneos</i>
fay	45	3,42333	0,780014	x
0	30	6,03333	1,03186	x
fre	45	8,46533	0,780014	x

<i>Contraste</i>	<i>Sig.</i>	<i>Diferencia</i>	<i>+/- Límites</i>
0 – fay	*	2,61	2,56479
0 – fre		-2,432	2,56479
fay – fre	*	-5,042	2,18726

\* indica una diferencia significativa.

Se deduce que el método de Fayyad & Irani requiere de muy poco tiempo en comparación con el método FRE –intervalos de igual frecuencia- y 0 –sin discretización- . Resalta que sin discretización se tiene mejor tiempo que utilizando método de discretización no supervisado FRE.

### Medias y 95,0% de Fisher LSD



## 6.5 Análisis de hipótesis

Como se ha comentado anteriormente, las hipótesis de partida para realizar correctamente el análisis de la varianza son:

- aleatoriedad de las muestras
- independencia de las variables
- normalidad de las distribuciones
- homogeneidad de las varianzas

La falta de normalidad de los datos, es quizás la violación de las hipótesis anteriores que menos influencia tiene sobre el contraste y las conclusiones extraídas del análisis sobre todo si los tamaños muestrales son suficientemente grandes, ya que las

medias siempre tendrán una distribución próxima a la normal según el Teorema Central del Límite.

Sin embargo, si las varianzas de todos los factores no son iguales entre sí, dará lugar a un aumento considerable del tamaño de la región crítica, con lo cual, el contraste siempre tenderá a rechazar la hipótesis nula. Se pueden verificar las hipótesis del modelo mediante un análisis pormenorizado de los residuos. Estos residuos son las cantidades que quedan después de eliminar las contribuciones sistemáticas del modelo propuesto. Si las hipótesis relativas al modelo son ciertas, se espera encontrar, aparte las restricciones impuestas por el análisis mismo, que los residuos varíen aleatoriamente. Por el contrario, si se descubre que los residuos contienen tendencias sistemáticas inexplicables, hay que sospechar del modelo. Por lo tanto, debe construirse y analizarse una tabla de los residuos como requisito imprescindible anterior a cualquier conclusión estadística.

Una forma para comprobar la suposición de normalidad consiste en hacer un histograma de los residuos. La suposición de que los errores se distribuyen idéntica e independientemente como variables normales de media cero y varianza  $\delta^2$  se simbolizará  $N(0, \delta^2)$ . Desafortunadamente, a menudo ocurren fluctuaciones considerables cuando el número de muestras es pequeño, por lo que una desviación moderada aparente de la normalidad no necesariamente implica una violación seria a las suposiciones.

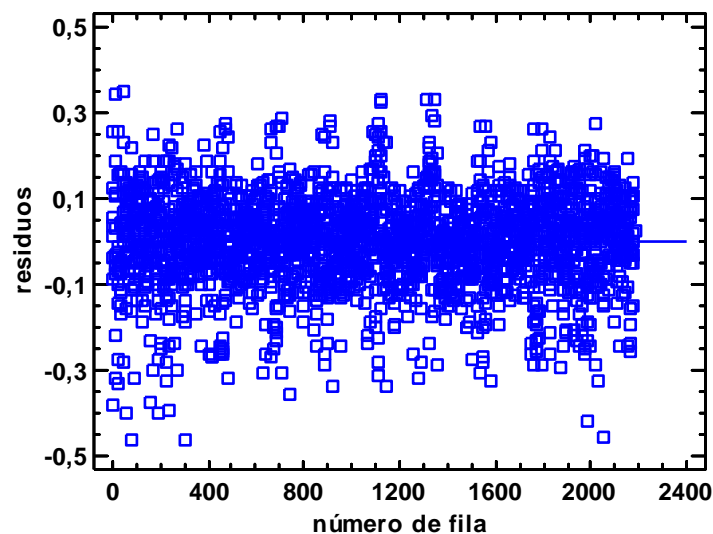
Un procedimiento gráfico para la verificación de esta hipótesis consiste en la construcción de una gráfica de probabilidad normal de los residuos. Una gráfica de este tipo es la representación de la distribución acumulada de los residuos sobre papel de probabilidad normal, es decir, es una gráfica cuya escala de ordenadas es tal que la distribución normal acumulada sea una recta. Para construir una gráfica de este tipo, deben disponerse los residuos en orden ascendente. Si la distribución de los errores es normal, esta gráfica parecerá una línea recta. Al visualizar dicha línea hay que poner mayor énfasis en los valores centrales de dicha gráfica, que en los extremos. Desviaciones grandes de la normalidad son potencialmente graves y requieren un análisis más profundo.

Si el modelo es correcto y las suposiciones se satisfacen, los residuos no deben seguir ningún patrón, ni deben estar relacionados con alguna otra variable, incluyendo

la respuesta. Una comprobación sencilla para verificar la homocedasticidad, consiste en representar los residuos frente los valores ajustados. En esta gráfica no debe revelarse ningún patrón obvio. Un defecto que en ocasiones revela la gráfica es el de una varianza variable. Algunas veces la varianza de las observaciones aumenta a medida que la magnitud de las observaciones lo hace. Esto puede suceder cuando el error es proporcional a la magnitud de la observación (comúnmente esto sucede en muchos instrumentos de medición, el error es proporcional a la escala de la lectura). Si la suposición de homocedasticidad no se cumple, el diseño está desbalanceado o si una varianza es mucho mayor que las otras, el problema es mucho más serio. La desigualdad en las varianzas del error puede perturbar significativamente las inferencias obtenidas sobre el análisis estadístico.

Una forma de ver de forma gráfica, aunque no precisa estas suposiciones es mediante el análisis de residuos:

**Gráfico de Residuos para accuracy**



Con estos residuos, se puede afirmar que se puede aplicar ANOVA.

## 6.6 Conclusión

Del estudio ANOVA podemos deducir para cada caso, resultados que nos permiten dar respuesta a las preguntas con las que abrimos esta tesis.

Sabemos que los sistemas de detección de intrusos que más triunfan en el mercado son los basados en uso indebido, los cuales necesitan contar con una base de datos o catálogo de firmas o patrones de ataque para su actualización periódica y así tener conocimiento de los nuevos ataques que se vayan desarrollando. Por otro lado los sistemas basados en anomalías se reservan al ámbito académico debido a que es una tecnología todavía no tan madura y no cuenta con la confianza de los fabricantes. Pero la gran ventaja con la que cuentan los sistemas basados en anomalías es que basándose en técnicas de inteligencia artificial como son las distintas técnicas de aprendizaje, éstos pueden aprender el comportamiento normal del sistema y modelar así, un perfil del mismo y toda desviación de ese modelo o perfil se considerará una anomalía o intrusión. Estos sistemas no necesitan de mantenimiento en forma actualizaciones periódicas sino sólo de un buen aprendizaje.

Basándonos en el conjunto de datos NSL-KDD se han llevado a cabo 3 estudios desde diferentes perspectivas sobre el mismo.

A todos los casos de estudio se les aplicó discretización, puesto que en el ámbito académico muchos trabajos hablan de la discretización como una buena técnica de reducción de datos y mejora el comportamiento de ciertos algoritmos, así como la selección de atributos. En total se realizaron 555 ejecuciones distintas.

En los 3 casos el estudio estadístico nos muestra que, tanto el método de discretización utilizado (supervisado, no supervisado y sin discretización) así como el método de selección de atributos (filtro, wrapper, y sin utilizar selección de atributos) y el algoritmo seleccionado, influyen en el error del clasificador y en el tiempo de construcción del modelo. El estudio ANOVA basándose en los datos de las tablas del apéndice A ofrece en cada caso conclusiones globales acerca de en qué medida estos 3 factores influyen en el comportamiento del modelo y en sus posteriores resultados.

En el primer caso los distintos ataques del sistema se mapearon a un solo tipo llamado “ataque” y en consecuencia el conjunto de datos se transformó en uno que sólo cuenta con dos tipos de conexiones: ataque y normal. ANOVA reveló que en el error

del clasificador, la discretización de tipo no supervisada –FRE, intervalos de igual frecuencia- mejora el acierto del sistema y el algoritmo del vecino más cercano con K=1 (KNN-1) es el que mejor resultado obtiene, seguido de los árboles de decisión, en concreto el RANDOM FOREST como se puede observar en las siguientes tablas:

Filtro	Discr	Algoritmo	TiempoTr	AcGlobal	AciertoA	AciertoN
all	0	KNN-1	0	99.83	99.67	100.00
fcfs	0	KNN-1	0	99.50	99.67	99.34
fcns	0	KNN-1	0	99.83	99.67	100.00
fc45	0	KNN-1	0	99.83	99.67	100.00
fnb	0	KNN-1	0	94.70	96.36	93.05
all	fay	KNN-1	0	99.23	98.81	100.00
fcfs	fay	KNN-1	0	95.65	96.44	94.20
fcns	fay	KNN-1	0	98.72	98.02	100.00
fc45	fay	KNN-1	0	97.70	98.02	97.10
fnb	fay	KNN-1	0	88.75	89.72	86.96
all	fre	KNN-1	0	99.83	99.64	100.00
fcfs	fre	KNN-1	0	99.48	99.64	99.33
fcns	fre	KNN-1	0	99.83	99.64	100.00
fc45	fre	KNN-1	0	99.83	99.64	100.00
fnb	fre	KNN-1	0	93.72	94.55	92.95

KNN-1 caso Binario

Filtro	Discr	Algoritmo	TiempoTr	AcGlobal	AciertoA	AciertoN
all	0	RandomForest	0.28	100.00	100.00	100.00
fcfs	0	RandomForest	0.09	99.34	100.00	98.68
fcns	0	RandomForest	0.08	100.00	100.00	100.00
fc45	0	RandomForest	0.08	100.00	100.00	100.00
fnb	0	RandomForest	0.08	94.21	96.03	92.38
all	fay	RandomForest	0.03	99.49	99.21	100.00
fcfs	fay	RandomForest	0.03	95.40	96.44	93.48
fcns	fay	RandomForest	0.05	98.47	98.02	99.28
fc45	fay	RandomForest	0.03	97.70	98.42	96.38
fnb	fay	RandomForest	0.02	88.75	89.72	86.96
all	fre	RandomForest	0.05	99.83	100.00	99.66
fcfs	fre	RandomForest	0.06	99.48	100.00	98.99
fcns	fre	RandomForest	0.05	100.00	100.00	100.00
fc45	fre	RandomForest	0.05	99.83	100.00	99.66
fnb	fre	RandomForest	0.05	93.72	94.91	92.62

Random Forest caso Binario

En cambio, la selección de atributos de tipo wrapper basada en naive bayes empeora su comportamiento y la selección de atributos basada en filtro CFS lo mejora. Pero si comparamos como influye CFS en el rendimiento del modelo en comparación



con el caso de no uso de la selección de atributos, esta técnica, CFS, tampoco ofrece una mejoría bastante patente. En este primer caso podemos afirmar que la selección de atributos no aporta ninguna notable mejoría en la tarea de clasificación del modelo.

En cuanto al tiempo que se requiere para la construcción y entrenamiento o aprendizaje del sistema, de nuevo la selección de características de tipo filtro CFS lo reduce considerablemente, y el algoritmo del vecino más cercano KNN-1 requiere de mucho menos tiempo, puesto que recordemos KNN no construye ningún modelo pertenece al paradigma perezoso, mientras que los modelos de MARKOV con distintos estados son los que más tiempo consumen, en concreto el HMM con 25 estados. Cabe resaltar que en el caso de los modelos de MARKOV se construye un HMM por cada categoría y se mide el tiempo de entrenamiento que tardan ambos HMM en ser entrenados. En cuanto al factor discretización cabe resalta que la discretización de tipo supervisada de Fayyad & Irani es la que requiere de menos tiempo y hace que la construcción y entrenamiento del modelo sea muy rápida.

En el segundo caso de estudio, el volumen de datos es mucho más grande y el sistema aprende a detectar entre ataques de 4 tipos distintos: Dos, Probe, R2L y U2R y una situación de actividad Normal, en total aprende a discernir entre 5 categorías. En este caso se utilizaron más tipos de algoritmos que en el primer y segundo caso con un total de 315 ejecuciones. Esto es debido a que en la literatura el caso al que los investigadores estudian más es el de clasificar entre estas 5 categorías.

ANOVA mostró que en el error del clasificador, el algoritmo que mejor comportamiento tiene es el RANDOM FOREST (bosque aleatorio, con un total de 10 árboles) y el de peor comportamiento es el algoritmo máquina de soporte vectorial con un kernel sigmoide, como se puede observar en las siguientes imágenes:

FILTRO	Discr	Algoritmo	TiempoTR	AcGlobal	AciertoD	AciertoN	AciertoP	AciertoR	AciertoU
all	0	RandomForest	0.0414	99.79	100.00	99.87	100.00	98.65	100.00
fcfs	0	RandomForest	2.42	98.94	99.96	99.65	99.65	93.16	100.00
fcns	0	RandomForest	3.76	99.77	100.00	99.91	100.00	98.43	100.00
fc45	0	RandomForest	3.95	99.75	99.96	99.78	100.00	98.65	100.00
fnb	0	RandomForest	2.56	92.39	90.44	98.10	95.89	78.59	75.38
all	fay	RandomForest	0.64	99.82	100.00	99.86	100.00	97.92	100.00
fcfs	fay	RandomForest	0.62	99.00	99.95	99.28	98.87	91.37	82.76
fcns	fay	RandomForest	0.23	99.82	100.00	99.86	100.00	97.92	100.00
fc45	fay	RandomForest	0.38	99.68	99.95	99.70	100.00	97.32	100.00
fnb	fay	RandomForest	0.23	94.77	94.28	97.57	92.12	68.45	62.07
all	fre	RandomForest	1.42	98.91	99.81	99.28	99.04	89.59	58.33
fcfs	fre	RandomForest	0.62	99.17	99.95	99.25	98.91	92.62	94.44
fcns	fre	RandomForest	0.76	99.86	99.97	99.92	100.00	98.05	100.00
fc45	fre	RandomForest	0.92	99.79	99.95	99.82	100.00	97.83	100.00
fnb	fre	RandomForest	0.66	95.60	95.40	97.96	92.08	73.97	75.00

Random Forest caso 5 categorías.

FILTRO	Discr	Algoritmo	TimeTr	AcGlobal	AciertoD	AciertoN	AciertoP	AciertoR	AciertoU
all	0	C-SVCSigmoide	286.1	47.04	45.70	96.41	0.00	0.34	0.00
fcfs	0	C-SVCSigmoide	257.37	32.67	0.00	100.00	0.28	0.00	0.00
fcns	0	C-SVCSigmoide	234.94	33.66	11.01	91.71	0.00	0.00	0.00
fc45	0	C-SVCSigmoide	684.56	44.56	53.10	80.95	0.21	0.34	0.00
fnb	0	C-SVCSigmoide	203.72	32.52	0.00	99.61	0.14	0.00	0.00
all	fay	C-SVCSigmoide	58.2	60.46	0.21	100.00	0.00	0.00	0.00
fcfs	fay	C-SVCSigmoide	18.66	60.40	0.00	100.00	0.00	0.00	0.00
fcns	fay	C-SVCSigmoide	29.03	60.40	0.00	100.00	0.00	0.00	0.00
fc45	fay	C-SVCSigmoide	31.59	60.40	0.00	100.00	0.00	0.00	0.00
fnb	fay	C-SVCSigmoide	15.63	47.37	93.09	99.93	0.00	0.00	0.00
all	fre	C-SVCSigmoide	105.09	54.38	0.00	100.00	0.00	0.00	0.00
fcfs	fre	C-SVCSigmoide	49.47	54.38	0.00	100.00	0.00	0.00	0.00
fcns	fre	C-SVCSigmoide	73.6	54.50	0.37	99.98	0.00	0.00	0.00
fc45	fre	C-SVCSigmoide	74.18	54.38	0.00	100.00	0.00	0.00	0.00
fnb	fre	C-SVCSigmoide	36.99	54.35	0.00	99.95	0.00	0.00	0.00

SVM Kernel Sigmoide caso 5 categorías.

La selección de atributos basada en filtro CFS ha sido la que mejor comportamiento ofrece seguida por la CNS que también es de tipo filtro. Por tanto en esta situación cabe decir que la selección de atributos basada en filtro es mejor que la basada en wrapper. Finalmente el método de discretización basado en entropía Fayyad & Irani, es el que mejor prestaciones ofrece seguido por la discretización no supervisada de intervalos de igual frecuencia y por último el no usar la discretización desmejora considerablemente las prestaciones del sistema.

En lo referente a tiempo, el algoritmo que requiere de más tiempo es la red neuronal perceptrón multicapa. La aplicación de la selección de características disminuye considerablemente el tiempo para la construcción del modelo en

comparación de su no uso, siendo la de tipo wrapper FNB basada en naive bayes la que más lo reduce seguida de la CFS. Sorprende que la discretización no reduzca el tiempo de construcción del modelo, esta situación se debe a que en los conjuntos discretizados se han evaluado con modelos ocultos de MARKOV los cuales requieren de más tiempo de entrenamiento puesto que se ha medido el tiempo total de entrenamiento de 5 modelos correspondientes a las 5 categorías.

En el último caso de estudio a nivel de detección de 19 ataques, y una situación normal, en total 20 categorías. ANOVA reveló que en el acierto de clasificación la selección de atributos no mejora las prestaciones, y el que peor comportamiento tiene es el FNB (wrapper basado en naive bayes) y el que mejor son los de tipo filtro, CFS seguido de CNS. El vecino más cercano KNN-1 es el que mejor prestaciones tiene seguido muy de cerca por RANDOM FOREST, siendo los modelos de MARKOV los peores.

FILTRO	Discr	Algoritmo	TiempoTR	AcGlobal
all	0	KNN-1	0	100.00
fcfs	0	KNN-1	0	99.10
fcns	0	KNN-1	0	100.00
fc45	0	KNN-1	0	99.74
fnb	0	KNN-1	0	83.59
all	fay	KNN-1	0	99.17
fcfs	fay	KNN-1	0	97.50
fcns	fay	KNN-1	0	97.92
fc45	fay	KNN-1	0	96.67
fnb	fay	KNN-1	0	80.00
all	fre	KNN-1	0	100.00
fcfs	fre	KNN-1	0	98.17
fcns	fre	KNN-1	0	99.83
fc45	fre	KNN-1	0	99.67
fnb	fre	KNN-1	0	80.83

KNN-1 20 categorías

FILTRO	Discr	Algoritmo	TiempoTR	AcGlobal
all	0	RandomForest	0.16	99.87
fcfs	0	RandomForest	0.08	98.97
fcns	0	RandomForest	0.13	95.51
fc45	0	RandomForest	0.16	99.62
fnb	0	RandomForest	0.55	83.46
all	fay	RandomForest	0.03	98.75
fcfs	fay	RandomForest	0.02	96.67
fcns	fay	RandomForest	0.03	96.67
fc45	fay	RandomForest	0.05	96.25
fnb	fay	RandomForest	0.05	78.33
all	fre	RandomForest	0.08	99.83
fcfs	fre	RandomForest	0.06	98.17
fcns	fre	RandomForest	0.08	99.33
fc45	fre	RandomForest	0.08	99.00
fnb	fre	RandomForest	0.05	80.33

Random Forest 20 categorías

La discretización de tipo no supervisada ha tenido mejores prestaciones, esto es debido a que el sistema ha contado con más datos para su entrenamiento que en la de Fayyad & Irani. En lo referente a tiempo cabe resaltar que en cuanto a algoritmo KNN-1 es el que menos tiempo requiere seguido por el árbol de decisión C4.5 y los que más tiempo necesitan son los modelos de MARKOV-se ha medido el tiempo total de entrenamiento de 20 HMM diferentes-. La selección de tipo CFS vuelve a ser la mejor en cuanto a reducción de tiempo y por último en lo que a discretización se refiere Fayyad es el que menos tiempo consume, seguido por el no uso de discretización y el que más tiempo necesita es la discretización supervisada.

Como conclusiones finales los algoritmos más sencillos como son el vecino más cercano con  $k=1$  y los árboles de decisión, especialmente RANDOM FOREST de Breiman, bosque con 10 árboles utilizando 6 atributos en la selección aleatoria, han dado muy buenos resultados frente al resto de algoritmos. Se trata de algoritmos sencillos, rápidos y ofrecen muy buenos resultados con casi el 100% de acierto de clasificación.

La discretización basada en Fayyad & Irani da buenos resultados frente a la no supervisada en grandes conjuntos de datos y ofrece mejores prestaciones en cuanto a clasificación y tiempo. La selección de atributos mejora el tiempo de entrenamiento del sistema, siendo la de tipo filtro la que mejor resultados ofrece y es más rápida que la de tipo wrapper y consume menos tiempo de computación a la hora ejecutarla para que nos ofrezca atributos relevantes. Cabe decir que en el caso de los modelos de MARKOV, al ser discretos sólo se han evaluado en conjuntos discretizados y han ofrecido peores prestaciones en tiempo debido a que durante la fase experimental el tiempo total es la suma del tiempo de entrenamiento de cada modelo de MARKOV que se crea por categoría a clasificar. En cuanto al acierto de clasificación los modelos de MARKOV rondan una media de acierto de más del 85%. Resalta que cuántos más números de estados se ha utilizado peor ha sido la prestación del modelo.

Por último se ha demostrado que aunque la tecnología de los sistemas de intrusos basados en anomalías no cuenta con el apoyo de los fabricantes, ésta es muy potente y llega a tener resultados excelentes utilizando técnicas de aprendizaje sencillas. Si se apostase por ella, aunque ofrece tasas de falsos positivos pero hemos visto en nuestro caso que pueden ser muy pequeñas casi nulas debido al elevado porcentaje de

acierto global del clasificador, el sistema tiene la capacidad de aprender y mejorar en muy poco tiempo, restando tiempo y posteriores tareas de mantenimiento periódico que necesitan los sistemas basados en uso indebido, puesto que identifican patrones de ataques y sus variantes que nunca han visto antes, teniendo definido previamente un perfil de comportamiento normal del sistema.

## 6.7 Trabajo Futuro

Este trabajo podría continuar con la evaluación de otras técnicas de inteligencia artificial y de técnicas de selección de atributos y discretización que no se han tenido en cuenta, así como la evaluación y comparativa de métodos híbridos, intentando mejorar la precisión de los modelos teniendo en cuenta las ventajas que ofrecen por separado.

Este trabajo está enfocado en el análisis de la eficiencia de diferentes paradigmas de cara a su posible implementación real, de dichos modelos en dispositivos programables, o programando un hardware de detección de intrusos basado en anomalías, o bien, para la implementación software en equipos y en redes para la protección de estos. Si permitimos que el modelo a implementar se adapte a los cambios que ocurren en la red, estaremos en vías de tener un Sistema de Detección de Intrusos actualizado y ajustado a los distintos tipos de ataques. Dicho trabajo se tendría que enfocar al análisis de los distintos protocolos de comunicación de red e incluso se podría encontrar sistemas de detección de intrusos que se ajusten a cada uno de esos protocolos.

## REFERENCIAS

---

- [Aha91] D. Aha, D. Kibler, and M. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991
- [All01] Allwein, E. L., Schapire, R. E. y Singer, Y. (Reducing multiclass to binary: a unifying approach for margin classifiers. *The Journal of Machine Learning Research*, 1:113-141. 2001
- [Amb03] Ambwani, T. Multi class support vector machine implementation to intrusion detection. *Proceedings of the International Joint Conference On Neural Networks*, Volume: 3, pp.2300–2305, 20-24 July 2003
- [Amo04] N. Ben Amor, S. Benferhat, Z. Elouedi. Naive Bayes vs decision trees in intrusion detection systems. *Proceedings of the 2004 ACM symposium on Applied Computing*. pp. 420-424. Nicosia, Cyprus.2004.
- [And80] James P. Anderson. *Computer Security Threat Monitoring and Surveillance*. Technical report, James P. Anderson Company, Fort Washington, Pennsylvania.
- [Anz92] Anzai, Y. *Pattern Recognition and Machine Learning*. Academic Press, Inc.1992
- [Asl95] T. Aslam, A Taxonomy of Security Faults in the UNIX Operating System, Purdue University Master's thesis, August 1995.
- [Att76] C.R. Attanasio, P.W. Markstein and R.J. Phillips, Penetrating an Operating System: A Study of VM/370 Integrity, *IBM System Journal*, vol. 15, 1, pp. 102-116, 1976.
- [Axe99b] S. Axelsson. The Base-Rate Fallacy and its Implications for the Difficulty of Intrusion Detection. In *6th ACM Conference on Computer and Communications Security*, 1999.
- [Axe00] Axelsson, S. *Intrusion Detection Systems: A Taxonomy and Survey*. Technical Report 99-15, Dept. of Computer Engineering, Chalmers University of Technology, Goteborg, Sweden.
- [Bar01] Daniel Barbara, NingningWu, and Sushil Jajodia. Detecting novel network intrusions using bayes estimators. In *Proceedings of First SIAM Conference on Data Mining*, Chicago, IL, 2001.
- [Bau67] Baum, L. E. y Egon, J. A. An inequality with applications to statistical estimation for probabilistic functions of a markov process and to a model for ecology. *BULLETIN OF THE AMERICAN METEOROLOGY SOCIETY*, 73:pp 360-363, 1967.

- [Bau72] Baum, L. E. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities*, 3:pp 1-8, 1972.
- [Bay73] T. Bayes. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions*, 53:370-418, 1763.
- [Bel73] D. E. Bell and J. L. LaPadula. Secure Computer Systems: Mathematical Foundations and Model. Technical Report M74-244, MITRE Corporation, Bedford, Massachusetts, May 1973.
- [Ben08] A. Ben-Hur, C. S. Ong, S. Sonnenburg, B. Scholkopf, and G. Ratsch. Support vector machines and kernels for computational biology. *PLoS Comp Biol*, 4(10):10-17, 2008.
- [Bib77] K. J. Biba. Integrity Constraints for Secure Computer Systems. Technical Report ESD-TR-76-372, USAF Electronic Systems Division, Bedford, Massachusetts, April 1977.
- [Bul92] A.L. Blum and P. Langley, Training 3-Node Neural Networks is NP-Complete, *Neural Networks*, Vol 5, pp. 117-127. 1992
- [Bou04] Bousquet, O., Boucheron, S. & Lugosi, G. (2004) Introduction to Statistical Learning Theory. en 'Advanced Lectures on Machine Learning' pp. 169-207.
- [Bre84] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and regression trees*. Wadsworth Int. Group, Belmont, CA, 1984.
- [Bre01] Leo Breiman. Random Forests. *Machine Learning*, 45, 5-32, 2001.
- [Bri00] Bridges, S. M. and Vaughn, R. B. Fuzzy Data Mining and Genetic Algorithms Applied to Intrusion Detection. In Proceedings of the 23rd National Information Systems Security Conference (NISSC 2000).
- [Bru04] Ferry Brugger. Data Mining Methods for Network Intrusion Detection. Thesis proposal, University of California, Davis. June 2004.
- [Cas02] L. Nunes de Castro and J. Timmis. An artificial immune network for multimodal function optimization. In Proceedings of the 2002 Congress on Evolutionary Computation (CEC'2002), volume 1, pp 669-674, Honolulu, Hawaii, May 2002.
- [Car89] Carbonell, J. G. 'Introduction: Paradigms for machine learning.' En J. G. Carbonell, editor, 'Machine Learning. Paradigms and methods,' Elsevier Science Publishers, Amsterdam, The Netherlands 1989

- [Cha03] P. Chan, M. Mahoney & M. Arshad. Learning Rules and Clusters for Anomaly Detection in Network Traffic. *Managing Cyber Threats: Issues, Approaches and Challenges*, V. Kumar, J. Srivastava & A. Lazarevic (editors), Kluwer, 2003.
- [Cl01] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [Coh95] William W. Cohen. Fast effective rule induction. In *In Proceedings of the Twelfth International Conference on Machine Learning*, pp 115–123. Morgan Kaufmann, 1995.
- [Coo92] Cooper, G. F. and E. Herskovits (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning* 9, 309–347.
- [Cos93] S. Cost and S. Salzberg. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, 10:57–78, 1993.
- [Cov67] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, IT-13(1):21–27, 1967.
- [Dar04] DARPA Intrusion Detection Evaluation, Lincoln Laboratory, Massachusetts Institute of Technology. [http://www.ll.mit.edu/IST/ideval/pubs/pubs\\_index.html](http://www.ll.mit.edu/IST/ideval/pubs/pubs_index.html), (07/10/2004).
- [Das99] Dipankar Dasgupta, editor. *Artificial Immune Systems and Their Applications*. Springer-Verlag, Berlin, 1999.
- [Das00] D. Dasgupta and F. Nino. A Comparison of Negative and Positive Selection Algorithms in Novel Pattern Detection. In the Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC), Volume: 1, Ppe(s): 125 -130, Nashville, October 8-11, 2000.
- [Das02a] D. Dasgupta and F. González. An Immunity-Based Technique to Characterize Intrusions in Computer Networks. *IEEE Transactions on Evolutionary Computation*, 6(3), pp 1081-1088, June 2002.
- [Das02b] D. Dasgupta and N.S. Majumdar. Anomaly Detection in Multidimensional Data Using Negative Selection Algorithm. In the proceedings of the Congress on Evolutionary Computation. WCCI 2002, Volume: 2, Ppe(s): 1039 -1044, Hawaii, May 14, 2002.
- [Deb99] H.Debar, M.Dacier and A.Wespi. A revised taxonomy for intrusion-detection systems. IBM Research Technical Report, October 1999.



- [Deb92a] H. Debar and Dorizzi, B. An Application of a Recurrent Network to an Intrusion Detection System. In IEEE, editor, International Joint Conference on Neural Networks 1992, pp 478-483.
- [Deb92b] H. Debar, M Becker, D. Siboni. A Neural Network Component for an Intrusion DetectionSystem. Proceedings, IEEE Symposium on Research in Computer Security and Privacy, 1992. pp 240-250.
- [Dem77] Dempster, A.P.; Laird, N.M.; y Rubin., D.B. Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society, B39(1):pp 1-38, 1977.
- [Den82] Dorothy E. Denning. Cryptography and Data Security. Addison-Wesley, Reading, Massachusetts, 1982.
- [Den87] Dorothy E. Denning. An Intrusion-Detection Model. IEEE transaction on Software Engineering, 13(2):222-232.
- [Dif76] Whitfield Diffie and Martin E. Hellman. New Directions in Cryptography. IEEE Transactions on Information Theory, vol. IT-22, num. 6, pp. 644-654, 1976.
- [Dou95] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In Proceedings of the 12<sup>th</sup> International Conference on Machine Learning, pp 194-202, Los Altos, CA, Morgan Kaufmann, 1995.
- [End98] D. Endler. Intrusion detection: Applying machine learning to solaris audit data. In Proceedings of the 1998 Annual Computer Security Applications Conference (ACSAC'98), pp 268--279, Los Alamitos, CA, December 1998. IEEE Computer Society, IEEE Computer Society Press. Scottsdale, AZ.
- [Ert03b] Levent Ertöz, Michael Steinbach, and Vipin Kumar. Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data. SIAM International Conference on Data Mining (SDM '03).
- [Esk02] Eskin, E., A. Arnold, M. Preraua, L. Portnoy, and S. J. Stolfo. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In D. Barbará and S. Jajodia (Eds.), Data Mining for Security Applications. Boston: Kluwer Academic Publishers. May 2002.
- [Fay93] U.M. Fayyad and K.B. Irani. Multi-interval discretization of continuous valued attributes for classification learning. In Proceedings of the 13th International Joint Conference on Artificial Intelligence, pp 1022-1027, Morgan Kaufmann, 1993.

- [Fay96] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. The KDD Process for Extracting Useful Knowledge from Volumes of Data. *Communications of the ACM* November 1996/Vol. 39, No. 11
- [Fix51] .E. Fix and J. Hodges. Discriminatory analysis, nonparametric discrimination consistency properties. Technical Report 4, US Air Force, School of Aviation Medicine, Randolph Field, TX, 1951
- [Fix52] E. Fix and J. Hodges. Discriminatory analysis, nonparametric discrimination: small sample performance. Technical Report 11, US Air Force, School of Aviation Medicine, Randolph Field, TX, 1952.
- [For89] Forsyth, R. 'The logic of induction.' En Chapman y H. Ltd., 'Machine Learning. Principles and Techniques,' Richard Forsyth. 1989.
- [For73] G. D. Forney, Jr. The viterbi algorithm. *PROC. IEEE (INVITED PAPER)*, 61:pp 268-278, Marzo 1973.
- [For94] S. Forrest, A. S. Perelson, L. Allen, and C. R. Cheru R. Kuri, Self-nonsel self discrimination in a computer, In *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, Los Alamitos, CA: IEEE Computer Society Press (1994).
- [For96] S. Forrest, S. Hofmeyr, A. Somayaji, and T. Longstaff, A sense of self for UNIX processes, in *Proc. IEEE Symposium on Computer Security and Privacy*, 1996.
- [Fox90] Fox, K., Henning, R., Reed, J., and Simonian, R. A Neural Network Approach Towards Intrusion Detection. *Proc. of the 13th National Computer Security Conference*, Washington, D.C., Oct. 1990, 125-134.
- [Fra98] Frank, E. Y Witten, I.H. (1998): "Generating Accurate Rule Sets Without Global Optimization", en J. SHAVLIK (ed.): *Proceedings of the Fifteenth International Conference on Machine Learning*, Madison, Wisconsin. Morgan Kaufmann, San Francisco, pp. 144-151
- [Fri97] Friedman, J. S., C. A. Tepley, P. A. Castleberg, and H. Roe, Middle-atmospheric Doppler lidar using an iodine-vapor edge filter, *Optics Letters*, 22, 1,648-1,650, 1997.
- [Fun89] Funahashi, K. On the approximate realization of continuous mapping by neural networks. *Neural Networks*, 2, 183-192. 1989.
- [Gao02] Bo Gao, Hui-Ye Ma, Yu-Hang Yang. HMMS (Hidden Markov Models) Based on Anomaly Intrusion Detection Method. *Proceedings of the First International Conference on Machine Learning and Cybernetics*, Beijing, 4-5 November 2002.

- [Gar96] S. Garfinkel, & G. Spafford, "Practical Unix & Internet Security", O'Reilly & Associates, Inc., 101 Morris Street, Sebastopol CA, 95472, 2nd edition, April 1996.
- [Gho99] Ghosh, A. and Schwartzbard, A. (1999a). A Study in Using Neural Networks for Anomaly and Misuse Detection. In Proceedings of the 8th USENIX Security Symposium (SEC'99).
- [Gol02] R. Goldman. A Stochastic Model for Intrusions. In Symposium on Recent Advances in Intrusion Detection (RAID), 2002.
- [Gol05] GOLDBERG, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley. 2005.
- [Gon02] F. González, D. Dasgupta, and R. Kozma. Combining Negative Selection and Classification Techniques for Anomaly Detection. In Proceedings of the Congress on Evolutionary Computation , pp 705-710, Honolulu, HI, May 2002. IEEE.
- [Gon03] F. González and D. Dasgupta. Anomaly detection using real-valued negative selection. Genetic Programming and Evolvable Machines, 4(4), pp 383-403, Kluwer Acad. Publ., December 2003.
- [Gua04] Jian Guan, Da-xin Liu and Tong Wang. Applications of Fuzzy Data Mining Methods for Intrusion Detection Systems. International Conference on Computational Science and Its Applications – ICCSA 2004:, Assisi, Italy, May 14-17, 2004, pp. 706 – 714.
- [Hal99] Hall, M. A. & Smith, L. A. Feature Selection for Machine Learning: Comparing a Correlation-Based Filter Approach to the Wrapper *in* 'Proceedings of the Twelfth International Florida Artificial Intelligence Research Society Conference, Orlando, USA' pp. 235-239. 1999
- [Hay94] Haykin S. Neural Networks, McMaster University , Ontario, Canada 1994.
- [Hea90] R. Heady, G. Luger, A. Maccabe, M. Servilla. The Architecture of a Network Level Intrusion Detection System. Technical report, Department of Computer Science, University of New Mexico, August 1990.
- [Heb90] Heberlein, L. T., Dias, G., Levitt, K., Mukherjee, B., Wood, J., and Wolber, D. A network security monitor. In Proceedings of the 1990 IEEE Computer Society Symposium on Research in Security and Privacy, pp 296-304.
- [Hel99] Helmer, G., Wong, J., Honavar, V., and Miller, L. (1999b). Feature selection using a genetic algorithm for intrusion detection (GECCO'99). In Banzhaf, W., Daida, J., Eiben, A. E.,

Garzon, M. H., Honavar, V., Jakiela, M., and Smith, R. E., editors, Proceedings of the Genetic and Evolutionary Computation Conference, ppe 1781.

[Hen94] Henery, R. J. Classification. Machine Learning, Neural and Statistical Classification, Michie, D., Spiegelhalter, D. J., and Taylor, C. C. (Eds.), Ellis Horwood, New York.

[Hol75] Holland J. H.: "Adaptation in Natural and Artificial Systems". Ann Arbor: The University of Michigan Press, 1975.

[Hol92] Holland J. H.: "Algoritmos Genéticos", revista Investigación y Ciencia, pág. 38-45. 1992.

[Hon97] S. J. Hong. Data Mining. Guest Editorial. Future Generation Computer Systems, Vol. 13, no. 2, pp. 95-97, Nov 1997.

[Hong02] Hong Han, Xian-Liang Lu, Li-Yong Ren. Using data mining to discover signatures in network-based intrusion detection. International Conference on Machine Learning and Cybernetics, 2002. Ppe(s): 13-17 vol.1.

[Hor89] Hornik, K., Stinchcombe, M. y White, H. Multilayer feedforward networks are universal approximators. Neural Networks, 2(5), 359-366. 1989

[Hüh09] Jens Christian Hühn and Eyke Hüllermeier. Furia: an algorithm for unordered fuzzy rule induction. *Data Mining and Knowledge Discovery*, 19(3):293–319, 2009.

[Hun66] E. Hunt, J. Marin, and P. Stone. *Experiments in induction*. Academic Press, New York, 1966.

[Int]<http://cert.inteco.es/cert/INTECOCERT/?jsessionid=1F07AA82BDE8C7A747C7504C3620A33D?postAction=getCertHome>

[Jac90] Jackson, K. A., Dubois, D. H., and Stallings, C. A. (1990). NADIR - A Prototype Network Intrusion Detection System. Technical Report LA-UR-90-3726, Los Alamos National Laboratory.

[Jay97] N.D. Jayaram and P.L.R. Morse, Network Security - A Taxonomic View, In Proceedings of the European Conference on Security and Detection, School of Computer Science, University of Westminster, UK, Publication No. 437, 28-30, April 1997.

[Jha01] S. Jha, K. Tan, and R. Maxion, Markov chains, Classifiers, and Intrusion Detection, Computer Security Foundations Workshop (CSFW), June 2001.

[Joh94] G. John, R. Kohavi, and K. Pflieger. Irrelevant features and the subset selection problem. In *11th Int. Conf. on Machine Learning*, pp 121–129, New Brunswick, NJ, 1994. Morgan Kaufmann.

[Kdd99] <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>

[Ken98] K. Kendall, A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems, Massachusetts Institute of Technology Master's Thesis, 1998.

[Koh96] R. Kohavi. Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pp 202-207, 1996.

[Koh97] R. Kohavi and G.H. John. Wrappers for feature Subset Selection. *Artificial Intelligence*, pp. 273-324. 1997

[Krs98] I. Krsul, Software Vulnerability Analysis, Purdue University Ph.D. dissertation, May 1998.

[Kru02] Christopher Kruegel, Thomas Toth and Engin Kirda. Service Specific Anomaly Detection for Network Intrusion Detection. In *Proceedings of the Symposium on Applied Computing (SAC)*, ACM Press. Spain, March 2002.

[Kru03] C. Kruegel, D. Mutz, W. Robertson, F. Valeur. Bayesian Event Classification for Intrusion Detection. 19th Annual Computer Security Applications Conference. Las Vegas, Nevada, December 08-12, 2003

[Kum94] Sandeep Kumar and Eugene Spafford. An Application of Pattern Matching in Intrusion Detection. Technical Report 94-013, Purdue University, Department of Computer Sciences, March 1994

[Kum95] Sandeep Kumar. Classification and Detection of Computer Intrusions. PhD thesis, Purdue University, West Lafayette, IN, USA, Aug 1995.

[Kum95b] S. Kumar, Classification and Detection of Computer Intrusion, Computer Science Department, Purdue University Ph.D. dissertation, August 1995.

[Lam71] B. W. Lampson. Protection. *Proceedings of the 5th Princeton Syrup. of Information Sci. and Syst.*, Princeton Univ., (1971), pp. 437-443

- [Land94] C. E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi, "A taxonomy of computer program security flaws," *ACM Comput. Surv.*, vol. 26, no. 3, pp. 211–254, 1994.
- [Lan94] P. Langley. Selection of relevant features in machine learning. In *Procs. Of the AAAI Fall Symposium on Relevance*, pp 140–144, 1994.
- [Lan96] Langley, P. *Elements of Machine Learning*. Morgan Kaufmann Publishers, Inc., San Francisco. 1996
- [Lane99] T. Lane, C. E. Brodley. Temporal Sequence Learning and Data Reduction for Anomaly Detection. *ACM Transactions on Information and System Security*, 2:295- 331, 1999.
- [Lan94] C. Landwehr, A. Bull, J. McDermott and W. Choi, A Taxonomy of Computer Program Security Flaws, *ACM Computing Surveys*, vol. 26, 3, pp. 211-254, September 1994.
- [Lee99] Wenke Lee, Sal Stolfo, and Kui Mok. Mining in a Data-flow Environment: Experience in Network Intrusion Detection. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '99)*, San Diego, CA, August 1999.
- [Li01] X. Li, and N. Ye. Decision tree classifiers for computer intrusion detection. *Journal of Parallel and Distributed Computing Practices*, Vol. 4, No. 2, 2001, pp. 179-190.
- [Lin94] Lin, T. *Fuzzy Patterns in Data*. In *17th National Computer Security Conference*. October 11-14, Baltimore, Maryland. 1994.
- [Lind97] U. Lindqvist and E. Jonsson, How to Systematically Classify Computer Security Intrusions, *IEEE Security and Privacy*, pp. 154-163, 1997.
- [Lip99] Lippmann, R. P. and Cunningham, R. K. (1999). Improving Intrusion Detection Performance using Keyword Selection and Neural Networks. *Web proceedings of the 2nd International Workshop on Recent Advances in Intrusion Detection (RAID'99)*.
- [Liu96] H. Liu and R. Setiono. A probabilistic approach to feature selection: A filter solution. In *Proceedings of the 13th International Conference on Machine Learning*, pp 319–327. Morgan Kaufmann, 1996.
- [Liu98] H. Liu and H. Motorola. *Feature Selection for Knowledge Discovery and Data Mining*. Boston: Kluwer Academy, 1998.
- [Liu05] H. Liu and Lei Yu. Toward Integrating Feature Selection Algorithms for Classification and Clustering. *IEEE Transactions on Knowledge and Data Engeneering*, Vol. 17, No. 4, April 2005

- [Lou01] D. Lough. A Taxonomy of Computer Attacks with Applications to Wireless Networks. Virginia Polytechnic Institute PhD Thesis, April 2001.
- [Lun88] Lunt, T. and Jagannathan, R. A Prototype Real-Time Intrusion-Detection Expert System. In Proceedings of the IEEE Symposium on Security and Privacy, pp 59- 66.
- [Lun90] Lunt, T. IDES: An intelligent System for Detecting Intruders. In Computer Security, Threats and Countermeasures.
- [Mar01] D. Marchette, Computer Intrusion Detection and Network Monitoring, A Statistical Viewpoint. New York, Springer, 2001.
- [Mé93] Ludovic Mé. Security Audit Trail Analysis Using Genetic Algorithms. In Proceedings of the 12th International Conference on Computer Safety, Reliability and Security, pp 329-340. 1993.
- [Mé96] Ludovic Mé. Genetic Algorithms, a Biologically Inspired Approach for Security Audit Trails Analysis. 1996 IEEE Symposium on Security and Privacy (SSP), Oakland (CA), may 1996.
- [Mé98] Ludovic Mé. GASSATA, A Genetic Algorithm as an Alternative Tool for Security Audit Trails Analysis. First international workshop on the Recent Advances in Intrusion Detection (RAID). September 14-16, 1998. Louvain-la-Neuve, Belgium.
- [Mit] MIT Lincoln Laboratory
- <http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/index.html>
- [Mit97] Mitchell, T. M. Machine Learning. McGraw-Hill. ISBN 0-07-042807-7. 1997.
- [Muk02] Mukkamala S., Janoski G., Sung A. H. Intrusion Detection Using Neural Networks and Support Vector Machines. Proceedings of IEEE International Joint Conference on Neural Networks, pp.1702-1707. 2002.
- [Muk04] S. Mukkamala, A. H. Sung and A. Abraham, Intrusion Detection Using Ensemble of Soft Computing and Hard Computing Paradigms, Journal of Network and Computer Applications, Elsevier Science, 2004.
- [Nae04] NAEIM, F., et al. Selection and Scaling of Ground Motion Time Histories for Structural Design Using Genetic Algorithms. En *Earthquake Spectra*. Vol. 20, No. 2 pp. 413-426. 2004

- [Nas98] M. Nassehi. Anomaly detection for Markov models. Technical Report Tech report RZ 3011 (#93057), IBM Research Division, Zurich Research Laboratory, March 1998.
- [Neu89a] P.G. Neumann and D.B. Parker. A Summary of Computer Misuse Techniques. In Proceedings of the 12th National Computer Security Conference, 396-407, 1989.
- [Neu89b] P.G. Neumann and D.B. Parker, COMPUTER CRIME Criminal Justice Resource Manual, U.S. Department of Justice National Institute of Justice Office of Justice Programs, Prepared by SRI International under contract to Abt Associates for National Institute of Justice, U.S. Department of Justice, contract #OJP-86-C-002., 1989.
- [Neu95] P.G. Neumann. Computer Related Risks. The ACM Press, a division of the Association for Computing Machinery, Inc. (ACM), 1995.
- [NSA89] National Security Agency. A Guide to Understanding Identification and Authentication in Trusted Systems NCSC-TG-017 Library No. 5-235,479 Version 1 [Light Blue Book]
- [NSL09] <http://www.iscx.ca/NSL-KDD/>
- [Orf03] Orfila, A.; Carbo, J.; Ribagorda, A.. Fuzzy logic on decision model for IDS. The 12<sup>th</sup> IEEE International Conference on Fuzzy Systems, 2003. FUZZ '03., Volume: 2 , 25-28 May 2003 Pp:1237 - 1242 vol.2.
- [Par75] D.B. Parker, Computer Abuse Perpetrators and Vulnerabilities of Computer Systems, Stanford Research Institute, Menlo Park, CA 94025 Technical Report, December 1975.
- [Pla98] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines, 1998.
- [Pll02] J.M. Peña, J.A. Lozano, and P. Larrañaga. Learning recursive Bayesian multinets for data clustering by means of constructive induction. *Machine Learning*, 47:63-89, 2002.
- [Port01] L. Portnoy, E. Eskin, S. Stolfo. Intrusion detection with unlabeled data using clustering. In ACM Workshop on Data Mining Applied to Security (DMSA) 2001.
- [Pow95] Richard Power. Current and Future Danger. Computer Security Institute, San Francisco, California, 1995.
- [Powe01] D. Powell and R. Stroud, Conceptual Model and Architecture, Deliverable D2, Project MAFTIA IST-1999-11583, IBM Zurich Research Laboratory Research Report RZ 3377, Nov. 2001.



- [Pre04] Prescher, Detlef. A tutorial on the expectation-maximization algorithm including maximum-likelihood estimation and em training of probabilistic context-free grammars, 2004. <http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0412015>.
- [Put02] R. Puttini, Z. Marrakchi, and L. Me. Bayesian Classification Model for Real-Time Intrusion Detection. In 22th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, 2002.
- [Qui86] Quinlan, J. R. 1986. Induction of decision trees. *Machine Learning*, 1, 81-106.
- [Qui93] J. R. Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann, San Mateo, California, 1993.
- [Rab90] Rabiner, Lawrence R. A tutorial on hidden markov models and selected applications in speech recognition. pp 267-296, 1990.
- [Ram03] M. Ramadas, S. Ostermann and B. Tjaden. Detecting Anomalous Network Traffic with Self-Organizing Maps. Web proceedings of the 6th International Workshop on Recent Advances in Intrusion Detection RAID, 2003
- [Ric99] T. Richardson, J. Davis, D. Jacobson, J. Dickerson and L. Elkin. Developing a Database of Vulnerabilities to Support the Study of Denial of Service Attacks. IEEE Symposium on Security and Privacy, May 1999.
- [Ric01] T. Richardson, The Development of a Database Taxonomy of Vulnerabilities to Support the Study of Denial of Service Attacks., Iowa State University PhD Thesis, 2001.
- [Rae04] Real Academia Española, Diccionario de la Lengua Española.
- [Rod88] Rodgers, J. L. & Nicewander, A. W. Thirteen Ways to Look at the Correlation Coefficient. *The American Statistician* 42: 59-66. 1988
- [Rum86] Rumelhart, D.E., Hinton, G.E. y Williams, R.J. Learning internal representations by error propagation. En: D.E. Rumelhart y J.L. McClelland (Eds.). *Parallel distributed processing* (pp. 318-362). Cambridge, MA: MIT Press. 1986
- [Rus91] Deborah Rusell and G. T. Gangemi Sr. *Computer Security Basics*. O'Reilly & Associates, Inc., Sebastopol, California, December 1991.
- [Rya98] Jake Ryan, Meng-Jang Lin, and Risto Miikkulainen. Intrusion Detection with Neural Networks. In *Advances in Neural Information Processing Systems 10 (Proceedings of NIPS'97, Denver, CO)*. Cambridge, MA: MIT Press, 1998

[Sch01] Schölkopf, B. y Smola, A. J. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA, USA 2001.

[Seby02] A. A. Sebyala, T. Olukemi, and L. Sacks. Active Platform Security through Intrusion Detection Using Naive Bayesian Network for Anomaly Detection. In London Communications Symposium, 2002.

[Sha49] Claude E. Shannon. Communication Theory of Secrecy Systems. Bell System Technical Journal, vol.28-4, ppe 656--715, 1949.

[Shaw04] Shawe-Taylor, J. y Cristianini, N. Kernel Methods for Pattern Analysis. Cambridge University Press, New York, NY, USA. 2004

[Shy03] M. Shyu, S. Chen, K. Sarinnapakorn, and L. Chang, "A novel anomaly detection scheme based on principal component classifier," Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM03), pp. 172–179, 2003.

[Spa89] Eugene Spafford. Crisis and Aftermath. Communications of the ACM, 32(6):678-687, June 1989.

[Sim83] Simon, H. A. 'Why should machines learn?' En R. S. Michalki, J. G. Carbonell

y T. M. Mitchell, editores, 'Machine Learning: An arti\_cial intelligence approach,' Tomo I. Morgan Kaufmann. 1983.

[Sta] Starlab, [users.pandora.be/Richard.wheeler1/ais/inn.html](http://users.pandora.be/Richard.wheeler1/ais/inn.html)

[Sto97] Sal Stolfo, Andreas Prodromidis, Shelley Tselepis, Wenke Lee, Dave Fan, and Phil Chan. JAM: Java Agents for Meta-learning over Distributed Databases. In Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD '97), Newport Beach, CA, August 1997

[Teng90] H. Teng, K. Chen, and S. Lu. Adaptive real-time anomaly detection using inductively generated sequential patterns. Proceedings of 1990 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, California, May 7-9, 1990, 278-84. Los Alamitos, CA: IEEE Computer Society Press.

- [Tim00] J. Timmis and M. Neal. Investigating the evolution and stability of a resource limited artificial immune system. In *Proc. of the Genetic and Evolutionary Computation Conference, Workshop on Artificial Immune System and Their Applications*, pp 40-41, 2000.
- [Utg97] Utgoff, P. E., Berkman, N. C. and Clouse, J. A. Decision Tree Induction Based on Efficient Tree Restructuring, *Machine Learning journal*, 10, pp. 5-44, 1997.
- [Val00] Alfonso Valdes and Keith Skinner. Adaptive, Model-based Monitoring for Cyber Attack Detection. *Recent Advances in Intrusion Detection (RAID 2000)*. Edited by H. Debar and L. Me and F. Wu. Toulouse, France. October, 2000. Pp 80–92.
- [Val01] A. Valdes and K. Skinner, Probabilistic Alert Correlation. In *Proceedings of the 4<sup>th</sup> International Symposium on Recent Advances in Intrusion Detection (RAID) 2001*. Lecture Notes in Computer Science, Number 2212. Springer-Verlag. 2001.
- [Vap95] Vapnik, V. N. *The Nature of Statistical Learning Theory*. Springer. 1995
- [War99] C. Warrender, S. Forrest, and B. Pearlmutter. Detecting intrusions using system calls: Alternative data models. In *IEEE Symposium on Security and Privacy*, pp 133–145, 1999.
- [Wei91] Weiss, S. M. y Kulikowski, C. A. *Computer Systems that Learn*. Morgan Kaufmann Publishers, Inc., San Francisco, CA. 1991
- [Wei04] Wei Li, Using Genetic Algorithm for Network Intrusion Detection. *Proceedings of the United States Department of Energy Cyber Security Group 2004 Training Conference*, Kansas City, Kansas, May 24-27, 2004.
- [Wein04] K. Q Weinberger, F. Sha, and L. K Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the twenty-first international conference on Machine learning*, 2004
- [Whi93] WHITLEY, D. A Genetic Algorithm Tutorial. Computer Science Department, Colorado State University, Technical Report CS-93-103, 1993.
- [Wit00] I. Witten and E. Frank. *Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann Publishers,(2000).
- [Ye00a] Nong Ye, Xiangyang Li and Syed Masum Emran. Decision Tree for Signature Recognition and State Classification. *IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop June 6-7, 2000 at West Point, New York*. pp. 194-199.

[Ye00b] Nong Ye and Xiangyang Li. Application of Decision Tree Classifier to Intrusion Detection (in press). In Proceedings of Second International Conference on DATA MINING 2000, Cambridge University, UK, July 2000.

[Yeu02] D. Y. Yeung, and C. Chow. Parzen-window Network Intrusion Detectors. Sixteenth International Conference on Pattern Recognition, Quebec City, Canada, August 2002, pp. 11-15.

[Zad65] Zadeh, L.A. Fuzzy sets. In Information and Control, 8: 338-352, 1965.

[Zhan03] Zhang Jian; Ding Yong; Gong Jian. Intrusion detection system based on fuzzy default logic. The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ '03., Volume: 2, 25-28 May 2003. Pp:1350-1356 vol.2

[Zhi03] Zhi-Song Pan, Song-Can Chen, Gen-Bao Hu, Dao-Giang Zhang. Hybrid neural network and C4.5 for misuse detection. Proceedings of the Second International Conference on Machine Learning and Cybernetics, Pp:2463 - 2467 Vol.4. Xian, 2-5 November 2003.



## ARTÍCULOS

---

**“State of Art of Intelligents Systems in E-Commerce”, First Spanish IT Conference, Conference Acts pages 381-385, Granada, September 2005.**

**“A Comparison of Decision Trees and SVM with and without applying Features Selections, for Classification of Intrusion-detection.” CEDI 2010 Intelligent System Symposium.**

**“A Comparison of Decision Trees and SVM with and without applying Features Selections, for Classification of Intrusion-detection.” Mathematical Models for Engineering Science. International Conference on Mathematical Models for Engineering Science MMES’10 Puerto de la Cruz , Tenerife November 30-December2, 2010.**

**“Statistical Study ANOVA for Different Artificial Intelligent Techniques applied to Intrusion Detection System for Binary Problem: Detecting Attacks and No attacks connections ”. En proceso de publicación**

**“Analysis of different Soft-Computing for Intrusion Detection System to Classify into 5 Categories: Dos, Probe, R2L, U2R and Normal” . En proceso de publicación.**

**“Statistical Study ANOVA for Different Artificial Intelligent Techniques applied to Intrusion Detection System to Detect Specific Attacks”. En proceso de publicación.**



# Apéndice A

## Tablas Filtro, Tiempo y Algoritmos





Primer Estudio A nivel de 2  
Categorías: Normal y Ataque



Etiqueta	Filtro	Discretizacion	Algoritmo	TiempoTrain	AciertoGlobal	AciertoAtaque	AciertoNormal
all	all	0	Furia	2.2	98.18	99.67	96.69
all	all	0	KNN-1	0	99.83	99.67	100.00
all	all	0	C4.5	0.2	98.34	99.67	97.02
all	all	0	RandomForest	0.28	100.00	100.00	100.00
all	all	0	SmoPoly	0.73	94.04	94.37	93.71
all	all	0	TAN	0.45	96.85	98.34	95.36
fcfs	fcfs	0	Furia	0.73	93.05	91.06	95.03
fcfs	fcfs	0	KNN-1	0	99.50	99.67	99.34
fcfs	fcfs	0	C4.5	0.0	99.17	100.00	98.34
fcfs	fcfs	0	RandomForest	0.09	99.34	100.00	98.68
fcfs	fcfs	0	SmoPoly	0.72	89.57	95.36	83.77
fcfs	fcfs	0	TAN	0.03	98.18	99.01	97.35
fcns	fcns	0	Furia	1.03	97.52	99.34	95.70
fcns	fcns	0	KNN-1	0	99.83	99.67	100.00
fcns	fcns	0	C4.5	0.05	98.51	99.67	97.35
fcns	fcns	0	RandomForest	0.08	100.00	100.00	100.00
fcns	fcns	0	SmoPoly	0.41	89.57	91.72	87.42
fcns	fcns	0	TAN	0.05	97.02	97.68	96.36
fc45	fc45	0	Furia	1.06	98.34	100.00	96.69
fc45	fc45	0	KNN-1	0	99.83	99.67	100.00
fc45	fc45	0	C4.5	0.06	98.68	100.00	97.35
fc45	fc45	0	RandomForest	0.08	100.00	100.00	100.00
fc45	fc45	0	SmoPoly	0.62	93.54	92.72	94.37
fc45	fc45	0	TAN	0.09	97.68	98.68	96.69
fnb	fnb	0	Furia	0.59	92.38	95.70	89.07
fnb	fnb	0	KNN-1	0	94.70	96.36	93.05
fnb	fnb	0	C4.5	0.02	92.38	96.69	88.08
fnb	fnb	0	RandomForest	0.08	94.21	96.03	92.38
fnb	fnb	0	SmoPoly	0.36	90.40	94.37	86.42
fnb	fnb	0	TAN	0.05	90.89	92.38	89.40
all	all	fay	Furia	0.42	95.91	96.84	94.20
all	all	fay	KNN-1	0	99.23	98.81	100.00
all	all	fay	C4.5	0.02	95.91	96.44	94.93
all	all	fay	RandomForest	0.03	99.49	99.21	100.00
all	all	fay	SmoPoly	0.19	95.91	97.23	93.48
all	all	fay	TAN	0.08	95.40	94.07	97.83
fcfs	fcfs	fay	Furia	0.19	93.61	97.63	86.23
fcfs	fcfs	fay	KNN-1	0	95.65	96.44	94.20
fcfs	fcfs	fay	C4.5	0.03	93.86	94.86	92.03
fcfs	fcfs	fay	RandomForest	0.03	95.40	96.44	93.48
fcfs	fcfs	fay	SmoPoly	0.13	93.09	95.26	89.13
fcfs	fcfs	fay	TAN	0.02	94.12	94.07	94.20
fcns	fcns	fay	Furia	0.42	96.93	98.02	94.93
fcns	fcns	fay	KNN-1	0	98.72	98.02	100.00
fcns	fcns	fay	C4.5	0.02	95.14	96.05	93.48
fcns	fcns	fay	RandomForest	0.05	98.47	98.02	99.28
fcns	fcns	fay	SmoPoly	0.16	82.35	83.40	80.43
fcns	fcns	fay	TAN	0.02	82.35	83.40	80.43
fc45	fc45	fay	Furia	0.28	94.63	96.05	92.03
fc45	fc45	fay	KNN-1	0	97.70	98.02	97.10
fc45	fc45	fay	C4.5	0.02	96.16	97.23	94.20
fc45	fc45	fay	RandomForest	0.03	97.70	98.42	96.38
fc45	fc45	fay	SmoPoly	0.19	94.12	94.47	93.48
fc45	fc45	fay	TAN	0.03	94.12	93.28	95.65
fnb	fnb	fay	Furia	0.14	86.70	84.58	90.58
fnb	fnb	fay	KNN-1	0	88.75	89.72	86.96
fnb	fnb	fay	C4.5	0.02	87.98	90.12	84.06
fnb	fnb	fay	RandomForest	0.02	88.75	89.72	86.96

fnb	fnb	fay	SmoPoly	0.08	89.51	88.93	90.58
fnb	fnb	fay	TAN	0.02	89.26	88.14	91.30
all	all	fre	Furia	0.84	94.42	99.64	89.60
all	all	fre	KNN-1	0	99.83	99.64	100.00
all	all	fre	C4.5	0.05	94.76	99.27	90.60
all	all	fre	RandomForest	0.05	99.83	100.00	99.66
all	all	fre	SmoPoly	1.69	100.00	100.00	100.00
all	all	fre	TAN	0.25	98.78	99.64	97.99
fcfs	fcfs	fre	Furia	0.8	93.37	100.00	87.25
fcfs	fcfs	fre	KNN-1	0	99.48	99.64	99.33
fcfs	fcfs	fre	C4.5	0.06	94.59	94.55	94.63
fcfs	fcfs	fre	RandomForest	0.06	99.48	100.00	98.99
fcfs	fcfs	fre	SmoPoly	0.78	97.21	99.64	94.97
fcfs	fcfs	fre	TAN	0.09	97.21	98.91	95.64
fcns	fcns	fre	Furia	0.58	93.19	100.00	86.91
fcns	fcns	fre	KNN-1	0	99.83	99.64	100.00
fcns	fcns	fre	C4.5	0.03	95.99	97.45	94.63
fcns	fcns	fre	RandomForest	0.05	100.00	100.00	100.00
fcns	fcns	fre	SmoPoly	1.17	99.30	100.00	98.66
fcns	fcns	fre	TAN	0.08	98.60	99.64	97.65
fc45	fc45	fre	Furia	0.83	93.02	100.00	86.58
fc45	fc45	fre	KNN-1	0	99.83	99.64	100.00
fc45	fc45	fre	C4.5	0.03	94.76	99.27	90.60
fc45	fc45	fre	RandomForest	0.05	99.83	100.00	99.66
fc45	fc45	fre	SmoPoly	1.11	99.48	100.00	98.99
fc45	fc45	fre	TAN	0.11	97.56	98.55	96.64
fnb	fnb	fre	Furia	1.08	87.78	94.91	81.21
fnb	fnb	fre	KNN-1	0	93.72	94.55	92.95
fnb	fnb	fre	C4.5	0.02	89.70	93.09	86.58
fnb	fnb	fre	RandomForest	0.05	93.72	94.91	92.62
fnb	fnb	fre	SmoPoly	0.45	90.92	94.18	87.92
fnb	fnb	fre	TAN	0.03	90.75	89.09	92.28
all	all	fay	Markov11	17.36	81.33	81.42	81.16
all	all	fay	Markov15	18.47	74.17	68.38	84.78
all	all	fay	Markov25	21.11	69.31	58.89	88.41
fcfs	fcfs	fay	Markov11	4.99	83.63	80.63	89.13
fcfs	fcfs	fay	Markov15	5.30	86.70	86.96	86.23
fcfs	fcfs	fay	Markov25	5.58	83.63	81.82	86.96
fcns	fcns	fay	Markov11	6.94	81.07	83.40	76.81
fcns	fcns	fay	Markov15	7.19	75.70	70.75	84.78
fcns	fcns	fay	Markov25	7.77	87.47	94.47	74.64
fc45	fc45	fay	Markov11	11.97	86.45	90.91	78.26
fc45	fc45	fay	Markov15	11.67	82.61	79.45	88.41
fc45	fc45	fay	Markov25	12.85	83.12	82.21	84.78
fnb	fnb	fay	Markov11	5.63	85.42	84.58	86.96
fnb	fnb	fay	Markov15	5.44	85.93	86.17	85.51
fnb	fnb	fay	Markov25	6.63	85.93	84.98	87.68
all	all	fre	Markov11	39.00	83.60	71.64	94.63
all	all	fre	Markov15	39.92	87.61	86.18	88.93
all	all	fre	Markov25	44.88	91.27	90.91	91.61
fcfs	fcfs	fre	Markov11	9.66	95.11	99.27	91.28
fcfs	fcfs	fre	Markov15	10.20	95.81	99.27	92.62
fcfs	fcfs	fre	Markov25	10.59	94.76	99.27	90.60
fcns	fcns	fre	Markov11	15.05	89.88	87.64	91.95
fcns	fcns	fre	Markov15	15.49	86.91	81.45	91.95
fcns	fcns	fre	Markov25	16.75	87.09	80.00	93.62
fc45	fc45	fre	Markov11	24.16	82.55	90.18	75.50
fc45	fc45	fre	Markov15	25.94	80.10	69.82	89.60
fc45	fc45	fre	Markov25	27.28	86.04	85.45	86.58

fnb	fnb	fre	Markov11	11.93	89.88	89.82	89.93
fnb	fnb	fre	Markov15	12.37	91.10	90.18	91.95
fnb	fnb	fre	Markov25	13.20	89.01	90.18	87.92

Segundo Estudio A nivel de 5  
Categorías: DoS, Probe, R2L, U2R y  
Normal





FILTRO	DISCRETIZACION	Algoritmo	TiempoConstruccionModelo	AciertoGlobal	AciertoD	AciertoN	AciertoP	AciertoR	AciertoU
all	0	Clonalg	25.82	40.26	63.41	20.73	60.00	0.00	0.00
all	0	Genetico	89.9	57.76	65.19	96.54	18.44	3.36	0.00
all	0	Furia	679.21	97.90	99.67	98.92	99.22	89.57	81.54
all	0	PART	21.83	99.21	99.75	99.65	99.50	96.41	95.38
all	0	RIPPER	112.27	98.53	99.92	99.40	98.87	92.15	96.92
all	0	RandomForest	0.0414	99.79	100.00	99.87	100.00	98.65	100.00
all	0	C4.5	10.58	98.99	99.75	99.57	99.36	95.85	84.62
all	0	NBTree	179.25	98.82	99.88	99.87	98.79	93.27	98.46
all	0	SimpleCart	115.47	98.45	99.67	99.31	99.08	93.16	81.54
all	0	SOMPoly	204.14	34.18	92.51	96.29	100.00	88.68	98.46
all	0	SOMRBF	2090.06	89.22	88.62	96.63	91.84	70.07	53.85
all	0	C-SVCRBF	59.33	92.35	90.73	96.20	92.62	89.13	53.85
all	0	C-SVCSigmoide	286.1	47.04	45.70	96.41	0.00	0.34	0.00
all	0	PML	3108.46	57.49	100.00	97.11	94.89	89.69	53.85
all	0	NaiveBayes	0.69	74.78	70.57	78.62	83.48	60.99	95.38
all	0	TAN	6.71	97.82	98.51	97.54	98.51	95.96	92.31
all	0	RBFNet5	157.65	78.56	79.55	95.64	74.68	41.70	23.08
all	0	KNN-1	0	99.73	99.96	100.00	99.93	98.09	100.00
all	0	KNN-50	0	94.14	95.90	96.98	93.62	86.21	47.69
fcfs	0	Clonalg	6.33	41.07	51.41	15.68	92.91	0.00	0.00
fcfs	0	Genetico	83.99	62.12	59.35	94.00	46.95	15.36	0.00
fcfs	0	Furia	409.08	96.66	99.83	99.22	97.23	82.62	67.69
fcfs	0	PART	5.1	97.87	99.92	99.44	97.09	90.47	84.62
fcfs	0	RIPPER	56.49	97.10	99.88	99.14	98.23	83.30	86.15
fcfs	0	RandomForest	2.42	98.94	99.96	99.65	99.65	93.16	100.00
fcfs	0	C4.5	1.14	97.89	99.83	99.44	97.66	89.91	84.62
fcfs	0	NBTree	14.83	97.49	99.79	99.27	96.74	88.79	84.62
fcfs	0	SimpleCart	48.43	98.18	99.92	99.35	98.44	91.03	84.62
fcfs	0	SOMPoly	617.93	86.26	87.00	95.59	86.67	60.76	67.69
fcfs	0	SOMRBF	2628	83.62	87.38	96.24	76.38	58.18	0.00
fcfs	0	C-SVCRBF	48.13	86.94	87.33	96.03	90.00	60.65	43.08
fcfs	0	C-SVCSigmoide	257.37	32.67	0.00	100.00	0.28	0.00	0.00
fcfs	0	PML	2255.49	88.88	86.63	96.98	89.86	75.00	53.85
fcfs	0	NaiveBayes	0.2	70.08	76.24	89.68	62.20	14.91	70.77
fcfs	0	TAN	0.75	96.44	97.97	97.75	96.67	89.46	83.08
fcfs	0	RBFNet5	21.09	80.05	87.00	95.33	78.58	26.01	50.77
fcfs	0	KNN-1	0	98.97	99.96	99.65	99.65	93.39	100.00
fcfs	0	KNN-50	0	94.07	98.39	96.29	90.50	85.76	46.15
fcns	0	Clonalg	7.83	37.26	55.30	20.65	58.94	0.00	0.00
fcns	0	Genetico	75.8	61.19	70.36	88.12	40.92	2.91	0.00
fcns	0	Furia	758.47	97.58	99.75	99.83	99.08	84.98	76.92
fcns	0	PART	7.27	99.18	99.88	99.65	99.29	96.08	96.92
fcns	0	RIPPER	70.44	98.08	99.75	99.18	98.30	91.59	81.54
fcns	0	RandomForest	3.76	99.77	100.00	99.91	100.00	98.43	100.00
fcns	0	C4.5	1.89	99.01	99.79	99.74	99.43	95.63	81.54
fcns	0	NBTree	36.94	98.68	99.71	99.74	99.15	93.05	89.23
fcns	0	SimpleCart	54.62	98.75	99.79	99.31	99.08	95.63	75.38
fcns	0	SOMPoly	287.17	87.49	78.56	94.90	91.84	87.44	61.54
fcns	0	SOMRBF	2005.51	82.16	78.60	96.54	87.16	52.58	0.00
fcns	0	C-SVCRBF	65.81	89.70	84.27	95.08	91.63	89.57	60.00
fcns	0	C-SVCSigmoide	234.94	33.66	11.01	91.71	0.00	0.00	0.00
fcns	0	PML	1975.41	95.29	96.07	95.68	97.30	91.48	61.54
fcns	0	NaiveBayes	0.23	65.99	66.02	78.23	64.96	33.86	92.31
fcns	0	TAN	1.44	97.21	97.85	97.24	98.23	95.18	78.46
fcns	0	RBFNet5	59.27	82.71	73.68	94.51	88.16	73.88	1.54
fcns	0	KNN-1	0	99.73	99.96	100.00	99.93	98.09	100.00
fcns	0	KNN-50	0	93.59	94.87	95.68	94.47	86.43	50.77
fc45	0	Clonalg	12.15	32.61	0.00	100.00	0.00	0.00	0.00
fc45	0	Genetico	79.39	56.09	67.92	93.13	12.13	0.00	20.00
fc45	0	Furia	638.12	96.84	99.42	99.14	98.30	83.07	76.92
fc45	0	PART	14.54	98.72	99.75	99.35	99.29	94.17	87.69
fc45	0	RIPPER	65.65	97.17	99.63	99.40	96.88	85.31	95.38
fc45	0	RandomForest	3.95	99.75	99.96	99.78	100.00	98.65	100.00
fc45	0	C4.5	4.57	98.41	99.46	99.35	98.58	93.61	87.69
fc45	0	NBTree	64.37	98.34	99.75	99.27	98.65	92.26	89.23
fc45	0	SimpleCart	68.04	98.53	99.79	98.96	99.08	94.62	78.46
fc45	0	SOMPoly	196.91	92.10	90.27	95.68	92.55	88.00	78.46
fc45	0	SOMRBF	1743.14	88.77	89.69	96.46	88.65	69.84	43.08
fc45	0	C-SVCRBF	49.48	92.00	90.11	95.98	92.20	89.13	55.38
fc45	0	C-SVCSigmoide	684.56	44.56	53.10	80.95	0.21	0.34	0.00
fc45	0	PML	3063.65	94.90	93.79	96.63	98.94	89.24	64.62
fc45	0	NaiveBayes	0.33	77.47	70.16	82.03	87.94	68.39	84.62
fc45	0	TAN	3.01	58.87	98.22	100.00	97.38	95.07	89.23
fc45	0	RBFNet5	272.85	87.08	90.56	90.71	80.85	80.94	47.69
fc45	0	KNN-1	0	99.66	99.92	99.87	99.93	97.98	100.00
fc45	0	KNN-50	0	94.28	96.27	96.63	94.40	85.99	47.69
fnb	0	Clonalg	7.57	50.65	93.63	57.58	0.00	0.00	0.00
fnb	0	Genetico	82.31	56.11	62.67	94.73	9.72	15.36	3.08
fnb	0	Furia	149.06	45.72	89.61	96.98	87.87	70.40	95.38
fnb	0	PART	3.87	90.66	90.02	97.80	94.47	69.51	67.69
fnb	0	RIPPER	19.34	88.17	89.74	97.80	81.99	69.96	70.77
fnb	0	RandomForest	2.56	92.39	90.44	98.10	95.89	78.59	75.38
fnb	0	C4.5	0.69	90.72	90.44	97.62	94.26	69.73	66.15
fnb	0	NBTree	7.61	89.69	90.07	98.19	89.93	68.61	56.92
fnb	0	SimpleCart	48.87	90.29	90.19	97.97	91.21	71.41	60.00
fnb	0	SOMPoly	2704.07	83.60	89.49	96.67	71.91	58.30	0.00

fnb	0	SOMRBF	3172.91	83.60	89.49	96.67	71.91	58.30	0.00
fnb	0	C-SVCRBF	44.91	86.53	89.78	96.85	82.84	59.98	43.08
fnb	0	C-SVCSigmoide	203.72	32.52	0.00	99.61	0.14	0.00	0.00
fnb	0	PML	2374.3	89.25	89.98	97.97	87.66	70.74	40.00
fnb	0	NaiveBayes	0.09	83.73	86.42	94.13	80.64	59.75	9.23
fnb	0	TAN	0.59	89.00	88.49	97.84	86.88	75.34	26.15
fnb	0	RBFNet5	505.54	83.28	82.16	96.89	80.99	56.73	53.85
fnb	0	KNN-1	0	46.52	90.15	98.49	95.39	75.78	98.46
fnb	0	KNN-50	0	88.19	89.74	97.02	84.04	70.96	43.08
all	fay	Clonalg	4.77	83.78	88.60	89.70	40.15	56.25	0.00
all	fay	Genetico	48.81	82.67	80.51	96.40	34.90	0.30	10.34
all	fay	Furia	85.72	98.39	99.85	99.47	97.94	79.46	68.97
all	fay	PART	1.64	98.98	99.79	99.45	97.94	91.96	75.86
all	fay	RiPPER	29.13	98.77	99.85	99.05	98.31	89.58	100.00
all	fay	RandomForest	0.64	99.82	100.00	99.86	100.00	97.92	100.00
all	fay	C4.5	0.44	98.58	99.64	99.38	97.00	86.90	72.41
all	fay	NBTree	42.7	98.51	99.74	98.94	98.31	89.88	55.17
all	fay	SimpleCart	107.39	98.97	99.74	99.51	99.25	89.88	65.52
all	fay	SOMPoly	45.55	99.47	99.95	99.63	99.62	94.35	100.00
all	fay	SOMRBF	532.53	97.68	99.90	98.91	96.81	76.79	24.14
all	fay	C-SVCRBF	67.31	60.55	100.00	99.65	99.62	92.56	79.31
all	fay	C-SVCSigmoide	58.2	60.46	0.21	100.00	0.00	0.00	0.00
all	fay	PML	39512.84	86.42	99.54	97.94	0.00	0.00	79.31
all	fay	NaiveBayes	0.03	90.24	89.22	90.00	96.81	89.88	79.31
all	fay	TAN	3.09	98.24	99.12	98.08	97.94	97.02	82.76
all	fay	RBFNet5	135.8	96.61	98.61	96.72	96.25	88.10	51.72
all	fay	KNN-1	0	99.80	99.90	100.00	100.00	96.43	100.00
all	fay	KNN-50	0	96.36	99.23	98.38	87.62	75.60	3.45
fcfs	fay	Clonalg	2.62	79.37	88.34	84.59	40.53	28.87	0.00
fcfs	fay	Genetico	45.8	82.26	83.29	96.12	12.76	15.18	0.00
fcfs	fay	Furia	79.66	97.43	99.79	98.80	93.62	76.19	51.72
fcfs	fay	PART	0.58	98.21	99.85	98.75	97.56	86.31	58.62
fcfs	fay	RiPPER	7.82	97.01	99.48	98.27	91.37	77.08	79.31
fcfs	fay	RandomForest	0.62	99.00	99.95	99.28	98.87	91.37	82.76
fcfs	fay	C4.5	0.14	98.38	99.69	99.05	96.06	88.99	62.07
fcfs	fay	NBTree	4.99	97.86	99.69	98.64	95.50	87.50	24.14
fcfs	fay	SimpleCart	41.47	98.65	99.79	99.12	97.56	89.88	72.41
fcfs	fay	SOMPoly	39.56	98.37	99.74	98.80	96.44	89.58	79.31
fcfs	fay	SOMRBF	334.48	96.65	99.43	98.24	94.00	72.62	0.00
fcfs	fay	C-SVCRBF	20.67	98.38	99.95	98.61	98.12	88.69	75.86
fcfs	fay	C-SVCSigmoide	18.66	60.40	0.00	100.00	0.00	0.00	0.00
fcfs	fay	PML	6007.43	97.49	99.85	98.57	97.19	78.87	0.00
fcfs	fay	NaiveBayes	0.03	93.73	94.89	94.45	91.56	88.10	13.79
fcfs	fay	TAN	0.22	97.03	95.72	98.59	94.75	89.88	75.86
fcfs	fay	RBFNet5	72.28	96.78	99.59	96.95	94.93	83.63	68.97
fcfs	fay	KNN-1	0	99.01	99.95	99.31	98.87	91.07	86.21
fcfs	fay	KNN-50	0	95.56	99.59	97.39	85.74	72.62	0.00
fcns	fay	Clonalg	3.53	81.45	78.44	92.54	49.91	13.10	0.00
fcns	fay	Genetico	46.84	86.53	81.79	96.37	63.23	28.87	31.03
fcns	fay	Furia	107.89	97.89	99.38	99.47	97.94	72.62	55.17
fcns	fay	PART	0.61	98.56	99.59	99.05	98.31	89.88	62.07
fcns	fay	RiPPER	16.19	98.20	99.54	98.73	98.31	87.20	55.17
fcns	fay	RandomForest	0.23	99.82	100.00	99.86	100.00	97.92	100.00
fcns	fay	C4.5	0.11	98.39	99.23	99.17	97.56	88.10	62.07
fcns	fay	NBTree	13.56	98.63	99.79	99.10	98.50	90.18	51.72
fcns	fay	SimpleCart	82.73	98.79	99.59	99.56	98.87	87.80	55.17
fcns	fay	SOMPoly	40.4	99.22	99.90	99.45	99.62	93.15	82.76
fcns	fay	SOMRBF	488.22	97.18	99.33	98.61	97.56	74.11	0.00
fcns	fay	C-SVCRBF	119.48	58.18	99.85	99.72	99.81	94.05	96.55
fcns	fay	C-SVCSigmoide	29.03	60.40	0.00	100.00	0.00	0.00	0.00
fcns	fay	PML	15532.48	94.31	99.79	98.50	96.25	13.69	0.00
fcns	fay	NaiveBayes	0.01	89.30	87.06	89.46	98.12	88.10	65.52
fcns	fay	TAN	0.7	97.99	98.56	98.08	98.12	96.13	65.52
fcns	fay	RBFNet5	71.36	96.47	97.01	96.97	96.25	92.26	37.93
fcns	fay	KNN-1	0	99.79	99.90	99.98	100.00	96.43	100.00
fcns	fay	KNN-50	0	95.92	97.11	98.45	91.93	71.13	0.00
fc45	fay	Clonalg	4.81	84.12	86.33	91.71	40.71	49.70	0.00
fc45	fay	Genetico	47.93	81.28	82.05	94.55	20.83	8.33	6.90
fc45	fay	Furia	64.16	97.95	99.48	99.31	95.50	77.68	72.41
fc45	fay	PART	1.2	98.24	99.54	98.82	96.44	88.69	68.97
fc45	fay	RiPPER	22.96	98.30	99.79	98.34	97.56	91.07	89.66
fc45	fay	RandomForest	0.38	99.68	99.95	99.70	100.00	97.32	100.00
fc45	fay	C4.5	0.28	98.17	99.69	98.89	95.50	87.50	62.07
fc45	fay	NBTree	24.45	98.34	99.85	99.03	94.37	88.69	79.31
fc45	fay	SimpleCart	77.47	98.76	99.48	99.12	98.50	91.37	86.21
fc45	fay	SOMPoly	48.56	98.56	99.79	98.73	97.00	92.26	93.10
fc45	fay	SOMRBF	422.87	97.06	99.79	98.80	94.18	71.73	0.00
fc45	fay	C-SVCRBF	46.27	98.91	99.59	99.38	98.69	91.37	75.86
fc45	fay	C-SVCSigmoide	31.59	60.40	0.00	100.00	0.00	0.00	0.00
fc45	fay	PML	11142.34	97.39	99.74	98.71	91.93	79.76	48.28
fc45	fay	NaiveBayes	0.03	88.28	82.52	90.04	96.25	88.99	55.17
fc45	fay	TAN	0.67	97.42	98.66	97.39	97.19	93.15	72.41
fc45	fay	RBFNet5	98.83	96.78	98.04	97.27	95.12	89.58	51.72
fc45	fay	KNN-1	0	99.67	99.95	99.86	99.81	95.24	100.00
fc45	fay	KNN-50	0	95.63	97.47	98.34	88.37	69.94	0.00
fnb	fay	Clonalg	2.7	83.31	85.46	95.56	7.50	40.48	0.00
fnb	fay	Genetico	52.46	82.90	80.61	95.15	38.46	15.18	10.34

fnb	fay	Furia	11.64	92.14	93.91	97.94	63.98	55.36	51.72
fnb	fay	PART	0.37	94.33	94.28	97.34	89.68	66.96	51.72
fnb	fay	RiPPER	3.26	92.32	93.35	97.18	66.98	67.26	55.17
fnb	fay	RandomForest	0.23	94.77	94.28	97.57	92.12	68.45	62.07
fnb	fay	C4.5	0.08	94.51	94.07	97.44	91.37	66.96	55.17
fnb	fay	NBTree	2.53	93.93	94.07	97.64	82.18	67.56	51.72
fnb	fay	SimpleCart	17.85	94.49	94.22	97.44	91.37	67.26	44.83
fnb	fay	SOMPoly	57.03	93.76	92.21	96.60	90.06	76.19	44.83
fnb	fay	SOMRBF	253.47	91.00	90.41	97.83	76.92	36.61	0.00
fnb	fay	C-SVCRBF	9.28	94.21	94.12	97.64	87.05	66.96	34.48
fnb	fay	C-SVCSigmoide	15.63	47.37	93.09	99.93	0.00	0.00	0.00
fnb	fay	PML	1158.99	94.54	94.48	97.27	90.81	69.64	48.28
fnb	fay	NaiveBayes	0.02	89.70	84.84	95.36	85.55	58.04	13.79
fnb	fay	TAN	0.13	93.86	93.60	97.87	81.61	67.26	44.83
fnb	fay	RBFNet5	188.32	93.29	93.45	95.91	86.68	72.92	48.28
fnb	fay	KNN-1	0	94.77	94.17	97.67	91.93	68.45	58.62
fnb	fay	KNN-50	0	91.17	92.99	96.79	68.29	52.38	0.00
all	fre	Clonalg	9.45	83.30	81.86	99.38	6.83	14.75	0.00
all	fre	Genetico	51.38	81.71	81.65	93.60	21.86	24.30	66.67
all	fre	Furia	312.13	61.46	99.92	99.51	98.63	76.57	97.22
all	fre	PART	2.45	98.85	99.84	99.16	96.58	92.19	75.00
all	fre	RiPPER	58.36	98.69	99.95	98.41	98.22	93.49	88.89
all	fre	RandomForest	1.42	98.91	99.81	99.28	99.04	89.59	58.33
all	fre	C4.5	0.48	98.41	99.81	98.69	96.58	91.11	33.33
all	fre	NBTree	67.37	98.91	99.81	99.28	99.04	89.59	58.33
all	fre	SimpleCart	321.22	99.33	99.92	99.55	99.18	93.28	80.56
all	fre	SOMPoly	101.93	99.59	99.97	99.72	99.86	94.36	100.00
all	fre	SOMRBF	1243.26	98.35	99.84	98.86	97.27	84.60	55.56
all	fre	C-SVCRBF	739.85	99.56	99.95	99.83	99.86	93.49	86.11
all	fre	C-SVCSigmoide	105.09	54.38	0.00	100.00	0.00	0.00	0.00
all	fre	PML	112184.44	54.38	0.00	100.00	0.00	0.00	0.00
all	fre	NaiveBayes	0.14	87.75	85.96	87.43	97.95	90.89	83.33
all	fre	TAN	3.42	98.75	99.02	99.00	98.77	94.14	88.89
all	fre	RBFNet5	153.61	94.65	93.65	96.27	96.99	80.91	58.33
all	fre	KNN-1	0	99.86	99.95	100.00	100.00	97.18	100.00
all	fre	KNN-50	0	96.88	99.58	97.64	87.16	84.82	38.89
fcfs	fre	Clonalg	4.12	82.11	81.73	99.25	0.00	0.00	0.00
fcfs	fre	Genetico	52.54	81.49	83.18	95.95	0.00	16.05	0.00
fcfs	fre	Furia	264.99	97.52	99.89	99.16	93.03	67.46	52.78
fcfs	fre	PART	1.76	60.76	100.00	99.00	97.54	87.64	69.44
fcfs	fre	RiPPER	16.04	97.51	99.76	97.91	97.40	75.27	80.56
fcfs	fre	RandomForest	0.62	99.17	99.95	99.25	98.91	92.62	94.44
fcfs	fre	C4.5	0.17	98.42	99.87	98.74	97.81	86.77	55.56
fcfs	fre	NBTree	8.07	98.16	99.68	98.61	96.99	88.29	13.89
fcfs	fre	SimpleCart	90.03	98.77	99.89	99.08	97.13	89.59	80.56
fcfs	fre	SOMPoly	67.3	98.37	99.74	98.59	96.86	87.85	83.33
fcfs	fre	SOMRBF	870.59	96.40	98.60	98.31	93.31	66.16	0.00
fcfs	fre	C-SVCRBF	115.3	98.79	99.89	98.91	98.63	89.80	80.56
fcfs	fre	C-SVCSigmoide	49.47	54.38	0.00	100.00	0.00	0.00	0.00
fcfs	fre	PML	11293.37	92.99	94.92	99.00	97.13	0.00	0.00
fcfs	fre	NaiveBayes	0.05	90.99	87.92	93.12	93.44	90.67	13.89
fcfs	fre	TAN	0.39	97.21	96.17	98.91	94.81	88.94	77.78
fcfs	fre	RBFNet5	128.47	96.25	98.23	96.80	91.94	82.43	61.11
fcfs	fre	KNN-1	0	99.15	99.95	99.30	98.50	92.19	94.44
fcfs	fre	KNN-50	0	96.09	99.95	96.79	84.70	80.91	2.78
fcns	fre	Clonalg	4.73	77.29	77.31	93.17	0.00	0.00	0.00
fcns	fre	Genetico	52.8	83.20	83.87	97.77	12.30	7.81	2.78
fcns	fre	Furia	241.72	98.38	99.84	99.46	99.32	73.97	58.33
fcns	fre	PART	1.58	98.74	99.29	99.36	96.99	90.89	75.00
fcns	fre	RiPPER	22.9	98.55	99.81	98.79	98.09	87.64	75.00
fcns	fre	RandomForest	0.76	99.86	99.97	99.92	100.00	98.05	100.00
fcns	fre	C4.5	0.31	98.73	99.47	99.21	97.95	89.37	75.00
fcns	fre	NBTree	25.6	98.96	99.87	99.30	99.04	90.02	61.11
fcns	fre	SimpleCart	186.95	99.30	99.79	99.72	98.50	92.41	83.33
fcns	fre	SOMPoly	85.29	99.30	99.95	99.55	99.32	91.11	94.44
fcns	fre	SOMRBF	1116.76	97.76	99.37	98.59	97.13	82.43	0.00
fcns	fre	C-SVCRBF	550.84	99.64	99.92	99.82	99.86	95.44	91.67
fcns	fre	C-SVCSigmoide	73.6	54.50	0.37	99.98	0.00	0.00	0.00
fcns	fre	PML	36217.24	86.90	0.00	98.19	97.95	0.00	38.89
fcns	fre	NaiveBayes	0.13	86.84	85.75	85.95	96.58	91.76	88.89
fcns	fre	TAN	0.75	98.42	99.15	98.48	98.22	94.14	69.44
fcns	fre	RBFNet5	105.46	95.36	95.80	95.45	97.40	89.59	66.67
fcns	fre	KNN-1	0	99.87	99.95	100.00	100.00	97.40	100.00
fcns	fre	KNN-50	0	97.42	98.94	98.07	93.44	87.20	41.67
fc45	fre	Clonalg	4.49	82.27	80.25	99.16	5.60	8.03	0.00
fc45	fre	Genetico	50.26	83.76	82.34	94.36	50.14	16.05	25.00
fc45	fre	Furia	139.45	97.97	99.92	98.59	96.31	78.09	77.78
fc45	fre	PART	2.01	98.33	99.63	99.00	94.54	87.20	72.22
fc45	fre	RiPPER	55.61	98.42	99.81	98.34	96.99	91.11	86.11
fc45	fre	RandomForest	0.92	99.79	99.95	99.82	100.00	97.83	100.00
fc45	fre	C4.5	0.36	98.24	99.81	98.69	94.26	88.29	66.67
fc45	fre	NBTree	45.19	98.41	99.74	98.83	93.72	90.24	88.89
fc45	fre	SimpleCart	268.9	99.14	99.74	99.41	97.81	93.49	91.67
fc45	fre	SOMPoly	108.41	99.09	99.84	99.16	98.50	93.06	97.22
fc45	fre	SOMRBF	1356.33	97.55	99.79	98.63	94.26	75.92	27.78
fc45	fre	C-SVCRBF	445.76	68.43	99.71	100.00	99.45	94.36	80.56
fc45	fre	C-SVCSigmoide	74.18	54.38	0.00	100.00	0.00	0.00	0.00

fc45	fre	PML	48240.63	97.35	99.74	97.54	97.27	83.08	0.00
fc45	fre	NaiveBayes	0.06	86.95	82.34	88.65	95.63	89.80	77.78
fc45	fre	TAN	1.09	98.11	99.10	98.26	97.54	90.02	83.33
fc45	fre	RBFNet5	153.36	96.08	98.12	95.88	96.04	83.95	69.44
fc45	fre	KNN-1	0	99.83	99.92	99.93	99.86	94.79	100.00
fc45	fre	KNN-50	0	96.50	99.10	97.35	89.62	81.13	19.44
fnb	fre	Clonalg	3.1	81.74	79.90	98.81	4.64	4.34	0.00
fnb	fre	Genetico	54.49	80.85	85.19	91.75	8.47	24.73	8.33
fnb	fre	Furia	35.21	92.91	95.00	97.37	64.75	65.51	55.56
fnb	fre	PART	0.84	94.83	95.61	97.61	89.21	64.86	50.00
fnb	fre	RIPPER	8.64	93.02	94.45	97.84	66.80	62.69	63.89
fnb	fre	RandomForest	0.66	95.60	95.40	97.96	92.08	73.97	75.00
fnb	fre	C4.5	0.12	95.14	95.51	97.52	89.75	71.80	69.44
fnb	fre	NBTree	5.46	94.73	95.32	97.34	86.20	71.80	66.67
fnb	fre	SimpleCart	44.48	94.91	95.56	97.67	87.98	68.11	52.78
fnb	fre	SOMPoly	140.91	94.73	95.45	96.97	88.25	73.10	55.56
fnb	fre	SOMRBF	827.83	92.58	93.10	97.29	78.96	56.18	0.00
fnb	fre	C-SVCRBF	37.52	94.85	95.08	97.77	89.34	67.90	41.67
fnb	fre	C-SVCSigmoide	36.99	54.35	0.00	99.95	0.00	0.00	0.00
fnb	fre	PML	4079.92	94.41	95.48	98.16	88.25	54.23	0.00
fnb	fre	NaiveBayes	0.05	90.11	87.10	95.55	85.11	57.48	25.00
fnb	fre	TAN	0.48	93.79	94.13	97.92	84.15	55.75	55.56
fnb	fre	RBFNet5	86.78	92.57	90.24	96.89	83.88	72.67	52.78
fnb	fre	KNN-1	0	95.66	95.45	98.19	91.67	72.67	72.22
fnb	fre	KNN-50	0	91.76	95.29	97.15	57.92	53.36	5.56
all	fay	Markov11	498.47	84.70	86.38	83.57	86.87	85.12	96.55
all	fay	Markov15	498.61	78.38	82.31	74.45	82.36	98.51	96.55
all	fay	Markov25	576.39	87.55	85.61	93.46	55.91	72.62	89.66
fcfs	fay	Markov11	129.73	90.73	94.22	88.52	92.87	94.94	100.00
fcfs	fay	Markov15	150.03	90.38	94.28	88.26	90.81	94.35	93.10
fcfs	fay	Markov25	163.46	90.79	94.53	88.63	91.74	94.64	100.00
fcns	fay	Markov11	185.95	86.59	88.81	84.63	89.49	94.05	89.66
fcns	fay	Markov15	181.35	87.23	86.64	86.37	90.24	96.13	96.55
fcns	fay	Markov25	211.47	88.63	93.30	86.65	83.30	94.94	96.55
fc45	fay	Markov11	316.67	81.14	79.16	81.89	83.86	78.57	82.76
fc45	fay	Markov15	302.27	84.35	78.39	87.62	93.25	62.20	89.66
fc45	fay	Markov25	167.05	77.18	78.29	74.82	85.55	89.58	58.62
fnb	fay	Markov11	140.86	83.11	82.62	84.43	75.80	80.95	79.31
fnb	fay	Markov15	144.88	83.24	88.14	80.75	87.05	83.04	58.62
fnb	fay	Markov25	157.90	83.41	88.14	81.31	83.68	84.82	58.62
all	fre	Markov11	763.57	84.49	83.87	86.22	78.55	76.14	88.89
all	fre	Markov15	785.23	81.00	85.11	76.98	81.42	98.05	88.89
all	fre	Markov25	823.39	84.92	85.33	83.66	87.57	93.71	86.11
fcfs	fre	Markov11	233.36	86.91	83.29	88.11	90.30	95.01	94.44
fcfs	fre	Markov15	214.29	88.08	86.75	87.93	91.26	95.23	97.22
fcfs	fre	Markov25	216.68	89.33	91.22	87.04	92.90	96.96	100.00
fcns	fre	Markov11	297.40	82.73	85.11	81.68	72.40	92.19	94.44
fcns	fre	Markov15	296.46	83.71	83.26	83.99	82.10	85.25	97.22
fcns	fre	Markov25	329.43	84.59	83.69	84.20	87.16	91.76	100.00
fc45	fre	Markov11	466.72	82.98	79.32	85.79	84.43	73.54	94.44
fc45	fre	Markov15	511.87	74.99	79.88	68.81	86.89	94.36	97.22
fc45	fre	Markov25	535.27	79.21	82.76	74.92	83.88	97.61	86.11
fnb	fre	Markov11	215.32	86.95	90.19	85.48	77.19	95.44	80.56
fnb	fre	Markov15	217.21	84.60	85.38	83.73	82.65	92.84	80.56
fnb	fre	Markov25	226.67	84.88	86.07	83.94	81.83	92.62	77.78



**Tercer Estudio a nivel de 20**  
**Categorías: a nivel de Ataques**







# Apéndice B

## Tablas de Matrices de Confusión



Matrices de Confusión para el Primer  
Estudio a nivel de 2 Categorías:  
Normal y Ataque



All					
Furia					
A	A	N	A	N	
A	301	1	A	99.67	0.33
N	10	292	N	3.31	96.69
KNN-1					
A	A	N	A	N	
A	301	1	A	99.67	0.33
N	0	302	N	0.00	100.00
C4.5					
A	A	N	A	N	
A	301	1	A	99.67	0.33
N	9	293	N	2.98	97.02
RandomF					
A	A	N	A	N	
A	302	0	A	100.00	0.00
N	0	302	N	0.00	100.00
SmoPoly					
A	A	N	A	N	
A	285	17	A	94.37	5.63
N	19	283	N	6.29	93.71
Tan					
A	A	N	A	N	
A	297	5	A	98.34	1.66
N	14	288	N	4.64	95.36

CNS					
Furia					
A	A	N	A	N	
A	300	2	A	99.34	0.66
N	13	289	N	4.30	95.70
KNN-1					
A	A	N	A	N	
A	301	1	A	99.67	0.33
N	0	302	N	0.00	100.00
C4.5					
A	A	N	A	N	
A	301	1	A	99.67	0.33
N	8	294	N	2.65	97.35
RandomF					
A	A	N	A	N	
A	302	0	A	100.00	0.00
N	0	302	N	0.00	100.00
SmoPoly					
A	A	N	A	N	
A	277	25	A	91.72	8.28
N	38	264	N	12.58	87.42
Tan					
A	A	N	A	N	
A	295	7	A	97.68	2.32
N	11	291	N	3.64	96.36

Fay					
Furia					
A	A	N	A	N	
A	238	15	A	94.07	5.93
N	3	135	N	2.17	97.83
KNN-1					
A	A	N	A	N	
A	250	3	A	98.81	1.19
N	0	138	N	0.00	100.00
C4.5					
A	A	N	A	N	
A	244	9	A	96.44	3.56
N	7	131	N	5.07	94.93
RandomF					
A	A	N	A	N	
A	251	2	A	99.21	0.79
N	0	138	N	0.00	100.00
SmoPoly					
A	A	N	A	N	
A	246	7	A	97.23	2.77
N	9	129	N	6.52	93.48
Tan					
A	A	N	A	N	
A	238	15	A	94.07	5.93
N	3	135	N	2.17	97.83

FayCNS					
Furia					
A	A	N	A	N	
A	248	5	A	98.02	1.98
N	7	131	N	5.07	94.93
KNN-1					
A	A	N	A	N	
A	248	5	A	98.02	1.98
N	0	138	N	0.00	100.00
C4.5					
A	A	N	A	N	
A	243	10	A	96.05	3.95
N	9	129	N	6.52	93.48
RandomF					
A	A	N	A	N	
A	248	5	A	98.02	1.98
N	1	137	N	0.72	99.28
SmoPoly					
A	A	N	A	N	
A	211	42	A	83.40	16.60
N	27	111	N	19.57	80.43
Tan					
A	A	N	A	N	
A	211	42	A	83.40	16.60
N	27	111	N	19.57	80.43

FrecAll					
Furia					
A	A	N	A	N	
A	274	1	A	99.64	0.36
N	31	267	N	10.40	89.60
KNN-1					
A	A	N	A	N	
A	274	1	A	99.64	0.36
N	0	298	N	0.00	100.00
C4.5					
A	A	N	A	N	
A	273	2	A	99.27	0.73
N	28	270	N	9.40	90.60
RandomF					
A	A	N	A	N	
A	275	0	A	100.00	0.00
N	1	297	N	0.34	99.66
SmoPoly					
A	A	N	A	N	
A	275	0	A	100.00	0.00
N	0	298	N	0.00	100.00
Tan					
A	A	N	A	N	
A	274	1	A	99.64	0.36
N	6	292	N	2.01	97.99

FrecCNS					
Furia					
A	A	N	A	N	
A	275	0	A	100.00	0.00
N	39	259	N	13.09	86.91
KNN-1					
A	A	N	A	N	
A	274	1	A	99.64	0.36
N	0	298	N	0.00	100.00

CFS					
Furia					
A	A	N	A	N	
A	275	27	A	91.06	8.94
N	15	287	N	4.97	95.03
KNN-1					
A	A	N	A	N	
A	301	1	A	99.67	0.33
N	2	300	N	0.66	99.34
C4.5					
A	A	N	A	N	
A	302	0	A	100.00	0.00
N	5	297	N	1.66	98.34
RandomF					
A	A	N	A	N	
A	302	0	A	100.00	0.00
N	4	298	N	1.32	98.68
SmoPoly					
A	A	N	A	N	
A	288	14	A	95.36	4.64
N	49	253	N	16.23	83.77
Tan					
A	A	N	A	N	
A	299	3	A	99.01	0.99
N	8	294	N	2.65	97.35

C4.5					
Furia					
A	A	N	A	N	
A	302	0	A	100.00	0.00
N	10	292	N	3.31	96.69
KNN-1					
A	A	N	A	N	
A	301	1	A	99.67	0.33
N	0	302	N	0.00	100.00
C4.5					
A	A	N	A	N	
A	302	0	A	100.00	0.00
N	8	294	N	2.65	97.35
RandomF					
A	A	N	A	N	
A	302	0	A	100.00	0.00
N	0	302	N	0.00	100.00
SmoPoly					
A	A	N	A	N	
A	280	22	A	92.72	7.28
N	17	285	N	5.63	94.37
Tan					
A	A	N	A	N	
A	298	4	A	98.68	1.32
N	10	292	N	3.31	96.69

FayCFS					
Furia					
A	A	N	A	N	
A	247	6	A	97.63	2.37
N	19	119	N	13.77	86.23
KNN-1					
A	A	N	A	N	
A	244	9	A	96.44	3.56
N	8	130	N	5.80	94.20
C4.5					
A	A	N	A	N	
A	240	13	A	94.86	5.14
N	11	127	N	7.97	92.03
RandomF					
A	A	N	A	N	
A	244	9	A	96.44	3.56
N	9	129	N	6.52	93.48
SmoPoly					
A	A	N	A	N	
A	241	12	A	95.26	4.74
N	15	123	N	10.87	89.13
Tan					
A	A	N	A	N	
A	238	15	A	94.07	5.93
N	8	130	N	5.80	94.20

FayC4.5					
Furia					
A	A	N	A	N	
A	243	10	A	96.05	3.95
N	11	127	N	7.97	92.03
KNN-1					
A	A	N	A	N	
A	248	5	A	98.02	1.98
N	4	134	N	2.90	97.10
C4.5					
A	A	N	A	N	
A	246	7	A	97.23	2.77
N	8	130	N	5.80	94.20
RandomF					
A	A	N	A	N	
A	249	4	A	98.42	1.58
N	5	133	N	3.62	96.38
SmoPoly					
A	A	N	A	N	
A	239	14	A	94.47	5.53
N	9	129	N	6.52	93.48
Tan					
A	A	N	A	N	
A	236	17	A	93.28	6.72
N	6	132	N	4.35	95.65

FrecCFS					
Furia					
A	A	N	A	N	
A	275	0	A	100.00	0.00
N	38	260	N	12.75	87.25
KNN-1					
A	A	N	A	N	
A	274	1	A	99.64	0.36
N	2	296	N	0.67	99.33
C4.5					
A	A	N	A	N	
A	260	15	A	94.55	5.45
N	16	282	N	5.37	94.63
RandomF					
A	A	N	A	N	
A	275	0	A	100.00	0.00
N	3	295	N	1.01	98.99
SmoPoly					
A	A	N	A	N	
A	274	1	A	99.64	0.36
N	15	283	N	5.03	94.97
Tan					
A	A	N	A	N	
A	272	3	A	98.91	1.09
N	13	285	N	4.36	95.64

FrecC4.5					
Furia					
A	A	N	A	N	
A	275	0	A	100.00	0.00
N	40	258	N	13.42	86.58
KNN-1					
A	A	N	A	N	
A	274	1	A	99.64	0.36
N	0	298	N	0.00	100.00

NB					
Furia					
A	A	N	A	N	
A	289	13	A	95.70	4.30
N	33	269	N	10.93	89.07
KNN-1					
A	A	N	A	N	
A	291	11	A	96.36	3.64
N	21	281	N	6.95	93.05
C4.5					
A	A	N	A	N	
A	292	10	A	96.69	3.31
N	36	266	N	11.92	88.08
RandomF					
A	A	N	A	N	
A	290	12	A	96.03	3.97
N	23	279	N	7.62	92.38
SmoPoly					
A	A	N	A	N	
A	285	17	A	94.37	5.63
N	41	261	N	13.58	86.42
Tan					
A	A	N	A	N	
A	279	23	A	92.38	7.62
N	32	270	N	10.60	89.40

FayNB					
Furia					
A	A	N	A	N	
A	214	39	A	84.58	15.42
N	13	125	N	9.42	90.58
KNN-1					
A	A	N	A	N	
A	227	26	A	89.72	10.28
N	18	120	N	13.04	86.96
C4.5					
A	A	N	A	N	
A	228	25	A	90.12	9.88
N	22	116	N	15.94	84.06
RandomF					
A	A	N	A	N	
A	227	26	A	89.72	10.28
N	18	120	N	13.04	86.96
SmoPoly					
A	A	N	A	N	
A	225	28	A	88.93	11.07
N	13	125	N	9.42	90.58
Tan					
A	A	N	A	N	
A	223	30	A	88.14	11.86
N	12	126	N	8.70	91.30

FrecNB					
Furia					
A	A	N	A	N	
A	261	14	A	94.91	5.09
N	56	242	N	18.79	8

**C4.5**

A	A	N	A	N
268	7	A	97.45	2.55
16	282	N	5.37	94.63

**RandomF**

A	A	N	A	N
275	0	A	100.00	0.00
0	298	N	0.00	100.00

**SmoPoly**

A	A	N	A	N
275	0	A	100.00	0.00
4	294	N	1.34	98.66

**Tan**

A	A	N	A	N
274	1	A	99.64	0.36
7	291	N	2.35	97.65

**Markov**

**Fay11**

A	A	N	A	N
206	47	A	81.42	18.58
26	112	N	18.84	81.16

**FayCFS11**

A	A	N	A	N
204	49	A	80.63	19.37
15	123	N	10.87	89.13

**FayCNS11**

A	A	N	A	N
211	42	A	83.40	16.60
32	106	N	23.19	76.81

**FayC4.511**

A	A	N	A	N
230	23	A	90.91	9.09
30	108	N	21.74	78.26

**FayNB11**

A	A	N	A	N
214	39	A	84.58	15.42
18	120	N	13.04	86.96

**Frec11**

A	A	N	A	N
197	78	A	71.64	28.36
16	282	N	5.37	94.63

**FrecCFS11**

A	A	N	A	N
273	2	A	99.27	0.73
26	272	N	8.72	91.28

**FrecCNS11**

A	A	N	A	N
241	34	A	87.64	12.36
24	274	N	8.05	91.95

**FrecC4.511**

A	A	N	A	N
248	27	A	90.18	9.82
73	225	N	24.50	75.50

**FrecNB11**

A	A	N	A	N
247	28	A	89.82	10.18
30	268	N	10.07	89.93

**C4.5**

A	A	N	A	N
273	2	A	99.27	0.73
28	270	N	9.40	90.60

**RandomF**

A	A	N	A	N
275	0	A	100.00	0.00
1	297	N	0.34	99.66

**SmoPoly**

A	A	N	A	N
275	0	A	100.00	0.00
3	295	N	1.01	98.99

**Tan**

A	A	N	A	N
271	4	A	98.55	1.45
10	288	N	3.36	96.64

**Fay15**

A	A	N	A	N
173	80	A	68.38	31.62
21	117	N	15.22	84.78

**FayCFS15**

A	A	N	A	N
220	33	A	86.96	13.04
19	119	N	13.77	86.23

**FayCNS15**

A	A	N	A	N
179	74	A	70.75	29.25
21	117	N	15.22	84.78

**FayC4.515**

A	A	N	A	N
201	52	A	79.45	20.55
16	122	N	11.59	88.41

**FayNB15**

A	A	N	A	N
218	35	A	86.17	13.83
20	118	N	14.49	85.51

**Frec15**

A	A	N	A	N
237	38	A	86.18	13.82
33	265	N	11.07	88.93

**FrecCFS15**

A	A	N	A	N
273	2	A	99.27	0.73
22	276	N	7.38	92.62

**FrecCNS15**

A	A	N	A	N
224	51	A	81.45	18.55
24	274	N	8.05	91.95

**FrecC4.515**

A	A	N	A	N
192	83	A	69.82	30.18
31	267	N	10.40	89.60

**FrecNB15**

A	A	N	A	N
248	27	A	90.18	9.82
24	274	N	8.05	91.95

**Fay25**

A	A	N	A	N
149	104	A	58.89	41.11
16	122	N	11.59	88.41

**FayCFS25**

A	A	N	A	N
207	46	A	81.82	18.18
18	120	N	13.04	86.96

**FayCNS25**

A	A	N	A	N
239	14	A	94.47	5.53
35	103	N	25.36	74.64

**FayC4.525**

A	A	N	A	N
208	45	A	82.21	17.79
21	117	N	15.22	84.78

**FayNB25**

A	A	N	A	N
215	38	A	84.98	15.02
17	121	N	12.32	87.68

**Frec25**

A	A	N	A	N
250	25	A	90.91	9.09
25	273	N	8.39	91.61

**FrecCFS25**

A	A	N	A	N
273	2	A	99.27	0.73
28	270	N	9.40	90.60

**FrecCNS25**

A	A	N	A	N
220	55	A	80.00	20.00
19	279	N	6.38	93.62

**FrecC4.525**

A	A	N	A	N
235	40	A	85.45	14.55
40	258	N	13.42	86.58

**FrecNB25**

A	A	N	A	N
248	27	A	90.18	9.82
36	262	N	12.08	87.92

Matrices de Confusión para el  
Segundo Estudio a nivel de 5  
Categorías: DoS, Probe, R2L, U2R y  
Normal





All

Clonalg						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1532	193	0	0	691	D	63.41	7.99	0.00	0.00	28.60
P		508	846	0	0	56	P	36.03	60.00	0.00	0.00	3.97
R		566	246	0	0	80	R	63.45	27.58	0.00	0.00	8.97
U		38	19	0	0	8	U	58.46	29.23	0.00	0.00	12.31
N		802	1033	0	0	480	N	34.64	44.62	0.00	0.00	20.73

Genetico						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1575	0	0	0	841	D	65.19	0.00	0.00	0.00	34.81
P		529	260	0	0	621	P	37.52	18.44	0.00	0.00	44.04
R		8	6	30	0	848	R	0.90	0.67	3.36	0.00	95.07
U		20	15	0	0	30	U	30.77	23.08	0.00	0.00	46.15
N		44	36	0	0	2235	N	1.90	1.56	0.00	0.00	96.54

Furia						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		2408	2	0	0	6	D	99.67	0.08	0.00	0.00	0.25
P		2	1399	1	0	8	P	0.14	99.22	0.07	0.00	0.57
R		0	2	799	2	89	R	0.00	0.22	89.57	0.22	9.98
U		0	0	2	53	10	U	0.00	0.00	3.08	81.54	15.38
N		5	13	7	0	2290	N	0.22	0.56	0.30	0.00	98.92

Part						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		2410	1	0	0	5	D	99.75	0.04	0.00	0.00	0.21
P		1	1403	1	0	5	P	0.07	99.50	0.07	0.00	0.35
R		0	3	860	0	29	R	0.00	0.34	96.41	0.00	3.25
U		0	1	0	62	2	U	0.00	1.54	0.00	95.38	3.08
N		2	1	4	1	2307	N	0.09	0.04	0.17	0.04	99.65

Ripper						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		2414	1	0	0	1	D	99.92	0.04	0.00	0.00	0.04
P		1	1394	2	1	12	P	0.07	98.87	0.14	0.07	0.85
R		1	2	822	4	63	R	0.11	0.22	92.15	0.45	7.06
U		0	0	0	63	2	U	0.00	0.00	0.00	96.92	3.08
N		3	2	8	1	2301	N	0.13	0.09	0.35	0.04	99.40

RNDF						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		2416	0	0	0	0	D	100.00	0.00	0.00	0.00	0.00
P		0	1410	0	0	0	P	0.00	100.00	0.00	0.00	0.00
R		0	0	880	0	12	R	0.00	0.00	98.65	0.00	1.35
U		0	0	0	65	0	U	0.00	0.00	0.00	100.00	0.00
N		1	0	2	0	2312	N	0.04	0.00	0.09	0.00	99.87

C4.5						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		2410	2	0	0	4	D	99.75	0.08	0.00	0.00	0.17
P		2	1401	1	1	5	P	0.14	99.36	0.07	0.07	0.35
R		0	0	855	0	37	R	0.00	0.00	95.85	0.00	4.15
U		0	1	2	55	7	U	0.00	1.54	3.08	84.62	10.77
N		3	4	3	0	2305	N	0.13	0.17	0.13	0.00	99.57

NBTree						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		2413	2	0	0	1	D	99.88	0.08	0.00	0.00	0.04
P		8	1393	0	0	9	P	0.57	98.79	0.00	0.00	0.64
R		0	0	832	2	58	R	0.00	0.00	93.27	0.22	6.50
U		0	0	0	64	1	U	0.00	0.00	0.00	98.46	1.54
N		3	0	0	0	2312	N	0.13	0.00	0.00	0.00	99.87

CART						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		2408	1	0	0	7	D	99.67	0.04	0.00	0.00	0.29
P		0	1397	1	0	12	P	0.00	99.08	0.07	0.00	0.85
R		0	1	831	2	58	R	0.00	0.11	93.16	0.22	6.50
U		0	0	3	53	9	U	0.00	0.00	4.62	81.54	13.85
N		2	5	9	0	2299	N	0.09	0.22	0.39	0.00	99.31

SMOPoly						Acierto					
	D	P	R	U	N		D	P	R	U	N

D	2235	2411	0	2347	179	D	31.16	33.62	0.00	32.72	2.50
P	9	1410	1	1393	80	P	0.31	48.74	0.03	48.15	2.77
R	1	885	791	881	95	R	0.04	33.36	29.82	33.21	3.58
U	0	63	3	64	14	U	0.00	43.75	2.08	44.44	9.72
N	16	2311	45	2223	2229	N	0.23	33.87	0.66	32.58	32.66

SMORBF						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		2141	7	13	0	255	D	88.62	0.29	0.54	0.00	10.55
P		28	1295	2	0	85	P	1.99	91.84	0.14	0.00	6.03
R		4	8	625	0	255	R	0.45	0.90	70.07	0.00	28.59
U		0	0	4	35	26	U	0.00	0.00	6.15	53.85	40.00
N		15	38	24	1	2237	N	0.65	1.64	1.04	0.04	96.63

CSV-RBF						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		2192	3	0	0	221	D	90.73	0.12	0.00	0.00	9.15
P		30	1306	2	1	71	P	2.13	92.62	0.14	0.07	5.04
R		3	10	795	0	84	R	0.34	1.12	89.13	0.00	9.42
U		0	0	4	35	26	U	0.00	0.00	6.15	53.85	40.00
N		12	25	50	1	2227	N	0.52	1.08	2.16	0.04	96.20

CSV-SIGM						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1104	0	198	0	1114	D	45.70	0.00	8.20	0.00	46.11
P		536	0	13	0	861	P	38.01	0.00	0.92	0.00	61.06
R		1	0	3	0	888	R	0.11	0.00	0.34	0.00	99.55
U		35	0	0	0	30	U	53.85	0.00	0.00	0.00	46.15
N		70	0	13	0	2232	N	3.02	0.00	0.56	0.00	96.41

PML						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		2416	0	0	0	135	D	94.71	0.00	0.00	0.00	5.29
P		1410	1338	1	0	63	P	50.14	47.58	0.04	0.00	2.24
R		891	3	800	0	88	R	50.00	0.17	44.89	0.00	4.94
U		65	0	20	35	10	U	50.00	0.00	15.38	26.92	7.69
N		2314	16	39	0	2248	N	50.12	0.35	0.84	0.00	48.69

NaiveBayes						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1705	168	22	22	499	D	70.57	6.95	0.91	0.91	20.65
P		23	1177	36	111	63	P	1.63	83.48	2.55	7.87	4.47
R		4	8	544	322	14	R	0.45	0.90	60.99	36.10	1.57
U		0	0	3	62	0	U	0.00	0.00	4.62	95.38	0.00
N		27	99	109	260	1820	N	1.17	4.28	4.71	11.23	78.62

TAN						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		2380	9	1	0	26	D	98.51	0.37	0.04	0.00	1.08
P		3	1389	1	0	17	P	0.21	98.51	0.07	0.00	1.21
R		1	0	856	6	29	R	0.11	0.00	95.96	0.67	3.25
U		0	0	1	60	4	U	0.00	0.00	1.54	92.31	6.15
N		4	26	26	1	2258	N	0.17	1.12	1.12	0.04	97.54

RBFNet						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1922	8	0	0	486	D	79.55	0.33	0.00	0.00	20.12
P		120	1053	0	0	237	P	8.51	74.68	0.00	0.00	16.81
R		2	10	372	0	508	R	0.22	1.12	41.70	0.00	56.95
U		0	10	3	15	37	U	0.00	15.38	4.62	23.08	56.92
N		26	34	41	0	2214	N	1.12	1.47	1.77	0.00	95.64

KNN-1						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		2415	0	0	0	1	D	99.96	0.00	0.00	0.00	0.04
P		0	1409	0	0	1	P	0.00	99.93	0.00	0.00	0.07
R		0	0	875	0	17	R	0.00	0.00	98.09	0.00	1.91
U		0	0	0	65	0	U	0.00	0.00	0.00	100.00	0.00
N		0	0	0	0	2315	N	0.00	0.00	0.00	0.00	100.00

KNN-50						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		2317	4	0	0	95	D	95.90	0.17	0.00	0.00	3.93
P		25	1320	0	2	63	P	1.77	93.62	0.00	0.14	4.47

R	6	8	769	0	109	R	0.67	0.90	86.21	0.00	12.22
U	1	6	5	31	22	U	1.54	9.23	7.69	47.69	33.85
N	19	20	31	0	2245	N	0.82	0.86	1.34	0.00	96.98

## FCFS

Clonalg							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1242	1170	0	0	4	D	51.41	48.43	0.00	0.00	0.17
P		98	1310	0	0	2	P	6.95	92.91	0.00	0.00	0.14
R		1	816	0	0	75	R	0.11	91.48	0.00	0.00	8.41
U		0	56	0	0	9	U	0.00	86.15	0.00	0.00	13.85
N		10	1942	0	0	363	N	0.43	83.89	0.00	0.00	15.68

Genetico							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1434	15	0	0	967	D	59.35	0.62	0.00	0.00	40.02
P		254	662	0	0	494	P	18.01	46.95	0.00	0.00	35.04
R		4	31	137	0	720	R	0.45	3.48	15.36	0.00	80.72
U		28	7	0	0	30	U	43.08	10.77	0.00	0.00	46.15
N		25	111	3	0	2176	N	1.08	4.79	0.13	0.00	94.00

Furia							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2412	2	0	0	2	D	99.83	0.08	0.00	0.00	0.08
P		11	1371	0	0	28	P	0.78	97.23	0.00	0.00	1.99
R		1	2	737	0	152	R	0.11	0.22	82.62	0.00	17.04
U		0	7	0	44	14	U	0.00	10.77	0.00	67.69	21.54
N		5	12	0	1	2297	N	0.22	0.52	0.00	0.04	99.22

Part							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2414	0	1	0	1	D	99.92	0.00	0.04	0.00	0.04
P		8	1369	3	0	30	P	0.57	97.09	0.21	0.00	2.13
R		0	0	807	3	82	R	0.00	0.00	90.47	0.34	9.19
U		0	0	1	55	9	U	0.00	0.00	1.54	84.62	13.85
N		3	4	5	1	2302	N	0.13	0.17	0.22	0.04	99.44

Ripper							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2413	2	0	0	1	D	99.88	0.08	0.00	0.00	0.04
P		1	1385	0	0	24	P	0.07	98.23	0.00	0.00	1.70
R		0	1	743	2	146	R	0.00	0.11	83.30	0.22	16.37
U		0	1	0	56	8	U	0.00	1.54	0.00	86.15	12.31
N		4	7	8	1	2295	N	0.17	0.30	0.35	0.04	99.14

RNDF							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2415	0	0	0	1	D	99.96	0.00	0.00	0.00	0.04
P		1	1405	0	0	4	P	0.07	99.65	0.00	0.00	0.28
R		0	0	831	0	61	R	0.00	0.00	93.16	0.00	6.84
U		0	0	0	65	0	U	0.00	0.00	0.00	100.00	0.00
N		3	1	3	1	2307	N	0.13	0.04	0.13	0.04	99.65

C4.5							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2412	3	0	0	1	D	99.83	0.12	0.00	0.00	0.04
P		11	1377	0	0	22	P	0.78	97.66	0.00	0.00	1.56
R		0	3	802	1	86	R	0.00	0.34	89.91	0.11	9.64
U		0	0	1	55	9	U	0.00	0.00	1.54	84.62	13.85
N		3	4	4	2	2302	N	0.13	0.17	0.17	0.09	99.44

NBTree							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2411	2	0	0	3	D	99.79	0.08	0.00	0.00	0.12
P		26	1364	0	0	20	P	1.84	96.74	0.00	0.00	1.42
R		0	2	792	7	91	R	0.00	0.22	88.79	0.78	10.20
U		0	0	3	55	7	U	0.00	0.00	4.62	84.62	10.77
N		4	7	5	1	2298	N	0.17	0.30	0.22	0.04	99.27

CART							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2414	1	0	0	1	D	99.92	0.04	0.00	0.00	0.04
P		2	1388	1	1	18	P	0.14	98.44	0.07	0.07	1.28
R		0	2	812	3	75	R	0.00	0.22	91.03	0.34	8.41
U		0	0	0	55	10	U	0.00	0.00	0.00	84.62	15.38
N		3	7	3	2	2300	N	0.13	0.30	0.13	0.09	99.35

SMOPoly							Acierto					
	D	P	R	U	N		D	P	R	U	N	

D	2102	3	0	0	311	D	87.00	0.12	0.00	0.00	12.87
P	73	1222	2	0	113	P	5.18	86.67	0.14	0.00	8.01
R	4	5	542	3	338	R	0.45	0.56	60.76	0.34	37.89
U	2	1	3	44	15	U	3.08	1.54	4.62	67.69	23.08
N	40	37	22	3	2213	N	1.73	1.60	0.95	0.13	95.59

SMORBF						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	2111	6	0	0	299	D	87.38	0.25	0.00	0.00	12.38
P	106	1077	1	0	226	P	7.52	76.38	0.07	0.00	16.03
R	17	3	519	0	353	R	1.91	0.34	58.18	0.00	39.57
U	11	35	3	0	16	U	16.92	53.85	4.62	0.00	24.62
N	38	31	18	0	2228	N	1.64	1.34	0.78	0.00	96.24

CSV-RBF						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	2110	11	0	0	295	D	87.33	0.46	0.00	0.00	12.21
P	58	1269	5	0	78	P	4.11	90.00	0.35	0.00	5.53
R	6	13	541	0	332	R	0.67	1.46	60.65	0.00	37.22
U	2	7	3	28	25	U	3.08	10.77	4.62	43.08	38.46
N	28	42	21	1	2223	N	1.21	1.81	0.91	0.04	96.03

CSV-SIGM						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	0	32	25	8	2351	D	0.00	1.32	1.03	0.33	97.31
P	0	4	1	0	1405	P	0.00	0.28	0.07	0.00	99.65
R	0	2	0	0	890	R	0.00	0.22	0.00	0.00	99.78
U	0	0	0	0	65	U	0.00	0.00	0.00	0.00	100.00
N	0	0	0	0	2315	N	0.00	0.00	0.00	0.00	100.00

PML						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	2093	1	0	0	322	D	86.63	0.04	0.00	0.00	13.33
P	26	1267	1	0	116	P	1.84	89.86	0.07	0.00	8.23
R	4	3	669	0	216	R	0.45	0.34	75.00	0.00	24.22
U	2	0	6	35	22	U	3.08	0.00	9.23	53.85	33.85
N	18	20	31	1	2245	N	0.78	0.86	1.34	0.04	96.98

NaiveBayes						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	1842	16	0	158	400	D	76.24	0.66	0.00	6.54	16.56
P	310	877	0	184	39	P	21.99	62.20	0.00	13.05	2.77
R	16	34	133	72	637	R	1.79	3.81	14.91	8.07	71.41
U	0	7	0	46	12	U	0.00	10.77	0.00	70.77	18.46
N	47	68	11	113	2076	N	2.03	2.94	0.48	4.88	89.68

TAN						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	2367	2	0	0	47	D	97.97	0.08	0.00	0.00	1.95
P	30	1363	1	0	16	P	2.13	96.67	0.07	0.00	1.13
R	1	1	798	2	90	R	0.11	0.11	89.46	0.22	10.09
U	0	0	0	54	11	U	0.00	0.00	0.00	83.08	16.92
N	11	18	20	3	2263	N	0.48	0.78	0.86	0.13	97.75

RBFNet						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	2102	20	0	0	294	D	87.00	0.83	0.00	0.00	12.17
P	162	1108	1	0	139	P	11.49	78.58	0.07	0.00	9.86
R	5	27	232	1	627	R	0.56	3.03	26.01	0.11	70.29
U	2	7	2	33	21	U	3.08	10.77	3.08	50.77	32.31
N	41	56	7	4	2207	N	1.77	2.42	0.30	0.17	95.33

KNN-1						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	2415	0	0	0	1	D	99.96	0.00	0.00	0.00	0.04
P	1	1405	0	0	4	P	0.07	99.65	0.00	0.00	0.28
R	0	0	833	0	59	R	0.00	0.00	93.39	0.00	6.61
U	0	0	0	65	0	U	0.00	0.00	0.00	100.00	0.00
N	3	1	3	1	2307	N	0.13	0.04	0.13	0.04	99.65

KNN-50						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	2377	12	0	0	27	D	98.39	0.50	0.00	0.00	1.12
P	66	1276	3	0	65	P	4.68	90.50	0.21	0.00	4.61
R	3	11	765	2	111	R	0.34	1.23	85.76	0.22	12.44

U	0	7	7	30	21	U	0.00	10.77	10.77	46.15	32.31
N	23	36	25	2	2229	N	0.99	1.56	1.08	0.09	96.29

## FCNS

Clonalg							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1336	403	0	0	677	D	55.30	16.68	0.00	0.00	28.02
P		523	831	0	0	56	P	37.09	58.94	0.00	0.00	3.97
R		118	693	0	0	81	R	13.23	77.69	0.00	0.00	9.08
U		3	51	0	0	11	U	4.62	78.46	0.00	0.00	16.92
N		102	1735	0	0	478	N	4.41	74.95	0.00	0.00	20.65

Genetico							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1700	14	22	0	680	D	70.36	0.58	0.91	0.00	28.15
P		366	577	12	8	447	P	25.96	40.92	0.85	0.57	31.70
R		3	28	26	0	835	R	0.34	3.14	2.91	0.00	93.61
U		0	7	0	0	58	U	0.00	10.77	0.00	0.00	89.23
N		177	77	20	1	2040	N	7.65	3.33	0.86	0.04	88.12

Furia							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2410	1	0	0	5	D	99.75	0.04	0.00	0.00	0.21
P		2	1397	1	0	10	P	0.14	99.08	0.07	0.00	0.71
R		0	2	758	1	131	R	0.00	0.22	84.98	0.11	14.69
U		0	0	0	50	15	U	0.00	0.00	0.00	76.92	23.08
N		3	0	1	0	2311	N	0.13	0.00	0.04	0.00	99.83

Part							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2413	0	0	0	3	D	99.88	0.00	0.00	0.00	0.12
P		0	1400	1	1	8	P	0.00	99.29	0.07	0.07	0.57
R		0	1	857	1	33	R	0.00	0.11	96.08	0.11	3.70
U		0	0	0	63	2	U	0.00	0.00	0.00	96.92	3.08
N		3	2	3	0	2307	N	0.13	0.09	0.13	0.00	99.65

Ripper							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2410	2	0	0	4	D	99.75	0.08	0.00	0.00	0.17
P		0	1386	1	0	23	P	0.00	98.30	0.07	0.00	1.63
R		1	0	817	2	72	R	0.11	0.00	91.59	0.22	8.07
U		0	0	0	53	12	U	0.00	0.00	0.00	81.54	18.46
N		4	3	11	1	2296	N	0.17	0.13	0.48	0.04	99.18

RNDF							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2416	0	0	0	0	D	100.00	0.00	0.00	0.00	0.00
P		0	1410	0	0	0	P	0.00	100.00	0.00	0.00	0.00
R		0	0	878	0	14	R	0.00	0.00	98.43	0.00	1.57
U		0	0	0	65	0	U	0.00	0.00	0.00	100.00	0.00
N		1	0	1	0	2313	N	0.04	0.00	0.04	0.00	99.91

C4.5							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2411	1	0	0	4	D	99.79	0.04	0.00	0.00	0.17
P		1	1402	1	0	6	P	0.07	99.43	0.07	0.00	0.43
R		0	0	853	0	39	R	0.00	0.00	95.63	0.00	4.37
U		0	1	0	53	11	U	0.00	1.54	0.00	81.54	16.92
N		3	1	2	0	2309	N	0.13	0.04	0.09	0.00	99.74

NBTree							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2409	3	0	0	4	D	99.71	0.12	0.00	0.00	0.17
P		6	1398	0	0	6	P	0.43	99.15	0.00	0.00	0.43
R		0	0	830	2	60	R	0.00	0.00	93.05	0.22	6.73
U		0	0	0	58	7	U	0.00	0.00	0.00	89.23	10.77
N		4	1	1	0	2309	N	0.17	0.04	0.04	0.00	99.74

Cart							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2411	2	0	0	3	D	99.79	0.08	0.00	0.00	0.12
P		1	1397	0	0	12	P	0.07	99.08	0.00	0.00	0.85
R		1	1	853	1	36	R	0.11	0.11	95.63	0.11	4.04
U		0	2	2	49	12	U	0.00	3.08	3.08	75.38	18.46
N		2	5	9	0	2299	N	0.09	0.22	0.39	0.00	99.31

SMOPoly							Acierto					
---------	--	--	--	--	--	--	---------	--	--	--	--	--

	D	P	R	U	N		D	P	R	U	N
D	1898	44	4	0	470	D	78.56	1.82	0.17	0.00	19.45
P	37	1295	5	2	71	P	2.62	91.84	0.35	0.14	5.04
R	5	2	780	2	103	R	0.56	0.22	87.44	0.22	11.55
U	0	1	6	40	18	U	0.00	1.54	9.23	61.54	27.69
N	17	35	58	8	2197	N	0.73	1.51	2.51	0.35	94.90

**SMORBF** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1899	33	0	0	484	D	78.60	1.37	0.00	0.00	20.03
P	96	1229	0	0	85	P	6.81	87.16	0.00	0.00	6.03
R	79	3	469	0	341	R	8.86	0.34	52.58	0.00	38.23
U	1	1	6	0	57	U	1.54	1.54	9.23	0.00	87.69
N	22	37	21	0	2235	N	0.95	1.60	0.91	0.00	96.54

**CSV-RBF** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	2036	15	28	0	337	D	84.27	0.62	1.16	0.00	13.95
P	42	1292	13	1	62	P	2.98	91.63	0.92	0.07	4.40
R	1	4	799	2	86	R	0.11	0.45	89.57	0.22	9.64
U	0	0	7	39	19	U	0.00	0.00	10.77	60.00	29.23
N	24	27	58	5	2201	N	1.04	1.17	2.51	0.22	95.08

**CSV-SIGM** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	266	0	0	10	2140	D	11.01	0.00	0.00	0.41	88.58
P	21	0	0	0	1389	P	1.49	0.00	0.00	0.00	98.51
R	372	0	0	2	518	R	41.70	0.00	0.00	0.22	58.07
U	12	0	0	0	53	U	18.46	0.00	0.00	0.00	81.54
N	183	0	0	9	2123	N	7.90	0.00	0.00	0.39	91.71

**PML** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	2321	6	4	0	85	D	96.07	0.25	0.17	0.00	3.52
P	13	1372	6	1	18	P	0.92	97.30	0.43	0.07	1.28
R	2	0	816	15	59	R	0.22	0.00	91.48	1.68	6.61
U	0	0	6	40	19	U	0.00	0.00	9.23	61.54	29.23
N	26	24	48	2	2215	N	1.12	1.04	2.07	0.09	95.68

**NaiveBayes** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1595	184	9	42	586	D	66.02	7.62	0.37	1.74	24.25
P	122	916	12	110	250	P	8.65	64.96	0.85	7.80	17.73
R	10	36	302	393	151	R	1.12	4.04	33.86	44.06	16.93
U	1	0	4	60	0	U	1.54	0.00	6.15	92.31	0.00
N	116	75	54	259	1811	N	5.01	3.24	2.33	11.19	78.23

**Tan** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	2364	2	0	0	50	D	97.85	0.08	0.00	0.00	2.07
P	7	1385	0	0	18	P	0.50	98.23	0.00	0.00	1.28
R	1	0	849	3	39	R	0.11	0.00	95.18	0.34	4.37
U	0	2	1	51	11	U	0.00	3.08	1.54	78.46	16.92
N	11	24	29	0	2251	N	0.48	1.04	1.25	0.00	97.24

**RBFNet** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1780	49	8	0	579	D	73.68	2.03	0.33	0.00	23.97
P	12	1243	3	0	152	P	0.85	88.16	0.21	0.00	10.78
R	1	2	659	0	230	R	0.11	0.22	73.88	0.00	25.78
U	0	0	5	1	59	U	0.00	0.00	7.69	1.54	90.77
N	11	56	60	0	2188	N	0.48	2.42	2.59	0.00	94.51

**KNN-1** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	2415	0	0	0	1	D	99.96	0.00	0.00	0.00	0.04
P	0	1409	0	0	1	P	0.00	99.93	0.00	0.00	0.07
R	0	0	875	0	17	R	0.00	0.00	98.09	0.00	1.91
U	0	0	0	65	0	U	0.00	0.00	0.00	100.00	0.00
N	0	0	0	0	2315	N	0.00	0.00	0.00	0.00	100.00

**KNN-50** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	2292	23	4	0	97	D	94.87	0.95	0.17	0.00	4.01
P	26	1332	12	2	38	P	1.84	94.47	0.85	0.14	2.70



R	0	3	771	3	115	R	0.00	0.34	86.43	0.34	12.89
U	0	0	7	33	25	U	0.00	0.00	10.77	50.77	38.46
N	32	26	37	5	2215	N	1.38	1.12	1.60	0.22	95.68

## FC4.5

Clonal							Acierto						
	D	P	R	U	N		D	P	R	U	N		
D		0	0	0	0	2416	D	0.00	0.00	0.00	0.00	100.00	
P		0	0	0	0	1410	P	0.00	0.00	0.00	0.00	100.00	
R		0	0	0	0	892	R	0.00	0.00	0.00	0.00	100.00	
U		0	0	0	0	65	U	0.00	0.00	0.00	0.00	100.00	
N		0	0	0	0	2315	N	0.00	0.00	0.00	0.00	100.00	

Genetico							Acierto						
	D	P	R	U	N		D	P	R	U	N		
D		1641	42	0	0	733	D	67.92	1.74	0.00	0.00	30.34	
P		757	171	0	72	410	P	53.69	12.13	0.00	5.11	29.08	
R		84	1	0	6	801	R	9.42	0.11	0.00	0.67	89.80	
U		35	0	0	13	17	U	53.85	0.00	0.00	20.00	26.15	
N		110	33	0	16	2156	N	4.75	1.43	0.00	0.69	93.13	

Furia							Acierto						
	D	P	R	U	N		D	P	R	U	N		
D		2402	0	0	0	14	D	99.42	0.00	0.00	0.00	0.58	
P		0	1386	0	0	24	P	0.00	98.30	0.00	0.00	1.70	
R		7	1	741	0	143	R	0.78	0.11	83.07	0.00	16.03	
U		0	0	0	50	15	U	0.00	0.00	0.00	76.92	23.08	
N		4	10	6	0	2295	N	0.17	0.43	0.26	0.00	99.14	

Part							Acierto						
	D	P	R	U	N		D	P	R	U	N		
D		2410	0	1	0	5	D	99.75	0.00	0.04	0.00	0.21	
P		0	1400	1	0	9	P	0.00	99.29	0.07	0.00	0.64	
R		1	1	840	2	48	R	0.11	0.11	94.17	0.22	5.38	
U		0	0	1	57	7	U	0.00	0.00	1.54	87.69	10.77	
N		3	4	7	1	2300	N	0.13	0.17	0.30	0.04	99.35	

Ripper							Acierto						
	D	P	R	U	N		D	P	R	U	N		
D		2407	5	0	0	4	D	99.63	0.21	0.00	0.00	0.17	
P		0	1366	1	0	43	P	0.00	96.88	0.07	0.00	3.05	
R		1	0	761	0	130	R	0.11	0.00	85.31	0.00	14.57	
U		0	0	0	62	3	U	0.00	0.00	0.00	95.38	4.62	
N		2	6	6	0	2301	N	0.09	0.26	0.26	0.00	99.40	

RNDF							Acierto						
	D	P	R	U	N		D	P	R	U	N		
D		2415	0	0	0	1	D	99.96	0.00	0.00	0.00	0.04	
P		0	1410	0	0	0	P	0.00	100.00	0.00	0.00	0.00	
R		0	0	880	0	12	R	0.00	0.00	98.65	0.00	1.35	
U		0	0	0	65	0	U	0.00	0.00	0.00	100.00	0.00	
N		1	1	3	0	2310	N	0.04	0.04	0.13	0.00	99.78	

C4.5							Acierto						
	D	P	R	U	N		D	P	R	U	N		
D		2403	2	1	0	10	D	99.46	0.08	0.04	0.00	0.41	
P		4	1390	1	0	15	P	0.28	98.58	0.07	0.00	1.06	
R		1	0	835	3	53	R	0.11	0.00	93.61	0.34	5.94	
U		0	1	0	57	7	U	0.00	1.54	0.00	87.69	10.77	
N		3	1	9	2	2300	N	0.13	0.04	0.39	0.09	99.35	

NBTree							Acierto						
	D	P	R	U	N		D	P	R	U	N		
D		2410	2	0	0	4	D	99.75	0.08	0.00	0.00	0.17	
P		8	1391	1	0	10	P	0.57	98.65	0.07	0.00	0.71	
R		1	0	823	2	66	R	0.11	0.00	92.26	0.22	7.40	
U		0	0	1	58	6	U	0.00	0.00	1.54	89.23	9.23	
N		3	6	7	1	2298	N	0.13	0.26	0.30	0.04	99.27	

Cart							Acierto						
	D	P	R	U	N		D	P	R	U	N		
D		2411	2	0	0	3	D	99.79	0.08	0.00	0.00	0.12	
P		1	1397	0	0	12	P	0.07	99.08	0.00	0.00	0.85	
R		0	0	844	4	44	R	0.00	0.00	94.62	0.45	4.93	
U		0	0	2	51	12	U	0.00	0.00	3.08	78.46	18.46	
N		2	7	13	2	2291	N	0.09	0.30	0.56	0.09	98.96	

SMOPoly							Acierto						
---------	--	--	--	--	--	--	---------	--	--	--	--	--	--

	D	P	R	U	N		D	P	R	U	N
D	2181	3	0	0	232	D	90.27	0.12	0.00	0.00	9.60
P	13	1305	0	0	92	P	0.92	92.55	0.00	0.00	6.52
R	1	9	785	4	93	R	0.11	1.01	88.00	0.45	10.43
U	0	0	3	51	11	U	0.00	0.00	4.62	78.46	16.92
N	15	35	46	4	2215	N	0.65	1.51	1.99	0.17	95.68

**SMORBF** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	2167	6	0	0	243	D	89.69	0.25	0.00	0.00	10.06
P	52	1250	1	0	107	P	3.69	88.65	0.07	0.00	7.59
R	4	10	623	0	255	R	0.45	1.12	69.84	0.00	28.59
U	0	5	4	28	28	U	0.00	7.69	6.15	43.08	43.08
N	18	40	23	1	2233	N	0.78	1.73	0.99	0.04	96.46

**CSV-RBF** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	2177	6	0	0	233	D	90.11	0.25	0.00	0.00	9.64
P	43	1300	1	1	65	P	3.05	92.20	0.07	0.07	4.61
R	3	13	795	0	81	R	0.34	1.46	89.13	0.00	9.08
U	0	0	3	36	26	U	0.00	0.00	4.62	55.38	40.00
N	16	28	48	1	2222	N	0.69	1.21	2.07	0.04	95.98

**CSV-SIGM** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1283	16	86	13	1018	D	53.10	0.66	3.56	0.54	42.14
P	968	3	4	0	435	P	68.65	0.21	0.28	0.00	30.85
R	515	0	3	0	374	R	57.74	0.00	0.34	0.00	41.93
U	53	0	0	0	12	U	81.54	0.00	0.00	0.00	18.46
N	432	0	9	0	1874	N	18.66	0.00	0.39	0.00	80.95

**PML** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	2266	2	1	0	147	D	93.79	0.08	0.04	0.00	6.08
P	0	1395	1	0	14	P	0.00	98.94	0.07	0.00	0.99
R	2	4	796	2	88	R	0.22	0.45	89.24	0.22	9.87
U	0	0	3	42	20	U	0.00	0.00	4.62	64.62	30.77
N	10	29	37	2	2237	N	0.43	1.25	1.60	0.09	96.63

**NaiveBayes** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1695	263	2	16	440	D	70.16	10.89	0.08	0.66	18.21
P	47	1240	17	93	13	P	3.33	87.94	1.21	6.60	0.92
R	8	28	610	100	146	R	0.90	3.14	68.39	11.21	16.37
U	1	1	8	55	0	U	1.54	1.54	12.31	84.62	0.00
N	34	177	65	140	1899	N	1.47	7.65	2.81	6.05	82.03

**TAN** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	2373	3	0	0	2416	D	49.52	0.06	0.00	0.00	50.42
P	6	1373	1	0	1410	P	0.22	49.21	0.04	0.00	50.54
R	0	0	848	8	887	R	0.00	0.00	48.65	0.46	50.89
U	0	0	1	58	65	U	0.00	0.00	0.81	46.77	52.42
N	9	30	30	2	2315	N	0.38	1.26	1.26	0.08	97.02

**RBFNet** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	2188	57	6	0	165	D	90.56	2.36	0.25	0.00	6.83
P	10	1140	0	0	260	P	0.71	80.85	0.00	0.00	18.44
R	2	30	722	1	137	R	0.22	3.36	80.94	0.11	15.36
U	0	4	3	31	27	U	0.00	6.15	4.62	47.69	41.54
N	84	61	70	0	2100	N	3.63	2.63	3.02	0.00	90.71

**KNN-1** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	2414	0	0	0	2	D	99.92	0.00	0.00	0.00	0.08
P	0	1409	0	0	1	P	0.00	99.93	0.00	0.00	0.07
R	0	0	874	0	18	R	0.00	0.00	97.98	0.00	2.02
U	0	0	0	65	0	U	0.00	0.00	0.00	100.00	0.00
N	0	1	2	0	2312	N	0.00	0.04	0.09	0.00	99.87

**KNN-50** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	2326	8	0	0	82	D	96.27	0.33	0.00	0.00	3.39
P	21	1331	2	0	56	P	1.49	94.40	0.14	0.00	3.97

R	5	12	767	0	108	R	0.56	1.35	85.99	0.00	12.11
U	0	7	5	31	22	U	0.00	10.77	7.69	47.69	33.85
N	19	29	30	0	2237	N	0.82	1.25	1.30	0.00	96.63

## FNB

Clonalg							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2262	0	0	0	154	D	93.63	0.00	0.00	0.00	6.37
P		744	0	0	0	666	P	52.77	0.00	0.00	0.00	47.23
R		532	0	0	0	360	R	59.64	0.00	0.00	0.00	40.36
U		42	0	0	0	23	U	64.62	0.00	0.00	0.00	35.38
N		982	0	0	0	1333	N	42.42	0.00	0.00	0.00	57.58

Genetico							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1514	5	0	62	835	D	62.67	0.21	0.00	2.57	34.56
P		759	137	0	0	514	P	53.83	9.72	0.00	0.00	36.45
R		7	13	137	0	735	R	0.78	1.46	15.36	0.00	82.40
U		35	0	0	2	28	U	53.85	0.00	0.00	3.08	43.08
N		64	55	3	0	2193	N	2.76	2.38	0.13	0.00	94.73

Furia							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2165	2	0	2342	249	D	45.50	0.04	0.00	49.22	5.23
P		59	1239	4	1326	108	P	2.16	45.29	0.15	48.46	3.95
R		5	8	628	859	250	R	0.29	0.46	35.89	49.09	14.29
U		1	6	4	62	12	U	1.18	7.06	4.71	72.94	14.12
N		21	27	21	2223	2245	N	0.46	0.60	0.46	49.00	49.48

Part							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2175	1	0	0	240	D	90.02	0.04	0.00	0.00	9.93
P		15	1332	4	0	59	P	1.06	94.47	0.28	0.00	4.18
R		2	9	620	2	259	R	0.22	1.01	69.51	0.22	29.04
U		3	6	4	44	8	U	4.62	9.23	6.15	67.69	12.31
N		12	23	15	1	2264	N	0.52	0.99	0.65	0.04	97.80

Ripper							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2168	2	0	0	246	D	89.74	0.08	0.00	0.00	10.18
P		8	1156	3	0	243	P	0.57	81.99	0.21	0.00	17.23
R		3	6	624	4	255	R	0.34	0.67	69.96	0.45	28.59
U		1	6	4	46	8	U	1.54	9.23	6.15	70.77	12.31
N		13	16	21	1	2264	N	0.56	0.69	0.91	0.04	97.80

RNDF							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2185	0	0	0	231	D	90.44	0.00	0.00	0.00	9.56
P		6	1352	4	0	48	P	0.43	95.89	0.28	0.00	3.40
R		2	8	701	0	181	R	0.22	0.90	78.59	0.00	20.29
U		1	6	4	49	5	U	1.54	9.23	6.15	75.38	7.69
N		8	18	18	0	2271	N	0.35	0.78	0.78	0.00	98.10

C4.5							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2185	4	0	0	227	D	90.44	0.17	0.00	0.00	9.40
P		14	1329	4	0	63	P	0.99	94.26	0.28	0.00	4.47
R		2	9	622	2	257	R	0.22	1.01	69.73	0.22	28.81
U		3	7	4	43	8	U	4.62	10.77	6.15	66.15	12.31
N		16	21	18	0	2260	N	0.69	0.91	0.78	0.00	97.62

NBTree							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2176	7	0	0	233	D	90.07	0.29	0.00	0.00	9.64
P		35	1268	5	0	102	P	2.48	89.93	0.35	0.00	7.23
R		3	8	612	2	267	R	0.34	0.90	68.61	0.22	29.93
U		2	6	4	37	16	U	3.08	9.23	6.15	56.92	24.62
N		15	14	13	0	2273	N	0.65	0.60	0.56	0.00	98.19

CART							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2179	3	0	0	234	D	90.19	0.12	0.00	0.00	9.69
P		14	1286	5	0	105	P	0.99	91.21	0.35	0.00	7.45
R		4	10	637	0	241	R	0.45	1.12	71.41	0.00	27.02
U		3	7	4	39	12	U	4.62	10.77	6.15	60.00	18.46
N		16	15	16	0	2268	N	0.69	0.65	0.69	0.00	97.97

SMOPoly							Acierto					
	D	P	R	U	N		D	P	R	U	N	

D	2162	4	0	0	250	D	89.49	0.17	0.00	0.00	10.35
P	155	1014	1	0	240	P	10.99	71.91	0.07	0.00	17.02
R	16	2	520	0	354	R	1.79	0.22	58.30	0.00	39.69
U	27	11	3	0	24	U	41.54	16.92	4.62	0.00	36.92
N	30	29	18	0	2238	N	1.30	1.25	0.78	0.00	96.67

SMORBF						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	2162	4	0	0	250	D	89.49	0.17	0.00	0.00	10.35
P	155	1014	1	0	240	P	10.99	71.91	0.07	0.00	17.02
R	16	2	520	0	354	R	1.79	0.22	58.30	0.00	39.69
U	27	11	3	0	24	U	41.54	16.92	4.62	0.00	36.92
N	30	29	18	0	2238	N	1.30	1.25	0.78	0.00	96.67

CSV-RBF						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	2169	7	0	0	240	D	89.78	0.29	0.00	0.00	9.93
P	123	1168	5	0	114	P	8.72	82.84	0.35	0.00	8.09
R	12	12	535	0	333	R	1.35	1.35	59.98	0.00	37.33
U	6	7	0	28	24	U	9.23	10.77	0.00	43.08	36.92
N	29	24	19	1	2242	N	1.25	1.04	0.82	0.04	96.85

CSV-SIGM						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	0	16	25	8	2367	D	0.00	0.66	1.03	0.33	97.97
P	0	2	2	0	1406	P	0.00	0.14	0.14	0.00	99.72
R	0	3	0	0	889	R	0.00	0.34	0.00	0.00	99.66
U	0	0	0	0	65	U	0.00	0.00	0.00	0.00	100.00
N	0	9	0	0	2306	N	0.00	0.39	0.00	0.00	99.61

PML						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	2174	0	1	0	241	D	89.98	0.00	0.04	0.00	9.98
P	23	1236	5	0	146	P	1.63	87.66	0.35	0.00	10.35
R	2	7	631	0	252	R	0.22	0.78	70.74	0.00	28.25
U	5	6	5	26	23	U	7.69	9.23	7.69	40.00	35.38
N	15	14	18	0	2268	N	0.65	0.60	0.78	0.00	97.97

NaiveBayes						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	2088	41	0	0	287	D	86.42	1.70	0.00	0.00	11.88
P	173	1137	3	0	97	P	12.27	80.64	0.21	0.00	6.88
R	6	28	533	1	324	R	0.67	3.14	59.75	0.11	36.32
U	5	35	2	6	17	U	7.69	53.85	3.08	9.23	26.15
N	25	93	18	0	2179	N	1.08	4.02	0.78	0.00	94.13

TAN						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	2138	2	0	0	276	D	88.49	0.08	0.00	0.00	11.42
P	30	1225	4	0	151	P	2.13	86.88	0.28	0.00	10.71
R	2	5	672	3	210	R	0.22	0.56	75.34	0.34	23.54
U	1	27	4	17	16	U	1.54	41.54	6.15	26.15	24.62
N	10	12	24	4	2265	N	0.43	0.52	1.04	0.17	97.84

RBFNet						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	1985	57	0	0	374	D	82.16	2.36	0.00	0.00	15.48
P	56	1142	1	0	211	P	3.97	80.99	0.07	0.00	14.96
R	2	21	506	1	362	R	0.22	2.35	56.73	0.11	40.58
U	0	7	1	35	22	U	0.00	10.77	1.54	53.85	33.85
N	20	33	19	0	2243	N	0.86	1.43	0.82	0.00	96.89

KNN-1						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	2178	0	0	2389	238	D	45.33	0.00	0.00	49.72	4.95
P	7	1345	3	1401	55	P	0.25	47.85	0.11	49.84	1.96
R	2	9	676	883	205	R	0.11	0.51	38.08	49.75	11.55
U	1	6	4	64	5	U	1.25	7.50	5.00	80.00	6.25
N	8	12	15	2278	2280	N	0.17	0.26	0.33	49.60	49.64

KNN-50						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	2168	7	0	0	241	D	89.74	0.29	0.00	0.00	9.98
P	106	1185	7	0	112	P	7.52	84.04	0.50	0.00	7.94
R	13	9	633	0	237	R	1.46	1.01	70.96	0.00	26.57

U	6	7	4	28	20	U	9.23	10.77	6.15	43.08	30.77
N	23	22	23	1	2246	N	0.99	0.95	0.99	0.04	97.02

Markov

Fay11						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1675	62	66	8	128	D	86.38	3.20	3.40	0.41	6.60
P		43	463	8	13	6	P	8.07	86.87	1.50	2.44	1.13
R		1	4	286	9	36	R	0.30	1.19	85.12	2.68	10.71
U		0	0	1	28	0	U	0.00	0.00	3.45	96.55	0.00
N		116	105	256	234	3617	N	2.68	2.43	5.91	5.41	83.57

Fay15						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1596	98	157	19	69	D	82.31	5.05	8.10	0.98	3.56
P		41	439	39	12	2	P	7.69	82.36	7.32	2.25	0.38
R		0	2	331	3	0	R	0.00	0.60	98.51	0.89	0.00
U		0	0	1	28	0	U	0.00	0.00	3.45	96.55	0.00
N		37	66	895	108	3222	N	0.85	1.52	20.68	2.50	74.45

Fay25						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1660	20	49	2	208	D	85.61	1.03	2.53	0.10	10.73
P		119	298	63	20	33	P	22.33	55.91	11.82	3.75	6.19
R		0	0	244	9	83	R	0.00	0.00	72.62	2.68	24.70
U		0	0	0	26	3	U	0.00	0.00	0.00	89.66	10.34
N		53	24	152	54	4045	N	1.22	0.55	3.51	1.25	93.46

FayCFS11						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1827	42	33	23	14	D	94.22	2.17	1.70	1.19	0.72
P		18	495	1	15	4	P	3.38	92.87	0.19	2.81	0.75
R		0	2	319	4	11	R	0.00	0.60	94.94	1.19	3.27
U		0	0	0	29	0	U	0.00	0.00	0.00	100.00	0.00
N		100	94	221	82	3831	N	2.31	2.17	5.11	1.89	88.52

FayCFS15						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1828	40	32	23	16	D	94.28	2.06	1.65	1.19	0.83
P		27	484	1	18	3	P	5.07	90.81	0.19	3.38	0.56
R		0	2	317	2	15	R	0.00	0.60	94.35	0.60	4.46
U		0	1	1	27	0	U	0.00	3.45	3.45	93.10	0.00
N		139	80	194	95	3820	N	3.21	1.85	4.48	2.20	88.26

FayCFS25						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1833	77	27	0	2	D	94.53	3.97	1.39	0.00	0.10
P		29	489	2	10	3	P	5.44	91.74	0.38	1.88	0.56
R		0	4	318	3	11	R	0.00	1.19	94.64	0.89	3.27
U		0	0	0	29	0	U	0.00	0.00	0.00	100.00	0.00
N		94	93	204	101	3836	N	2.17	2.15	4.71	2.33	88.63

FayCNS11						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1722	98	38	1	80	D	88.81	5.05	1.96	0.05	4.13
P		47	477	5	1	3	P	8.82	89.49	0.94	0.19	0.56
R		0	1	316	3	16	R	0.00	0.30	94.05	0.89	4.76
U		0	0	1	26	2	U	0.00	0.00	3.45	89.66	6.90
N		56	104	362	143	3663	N	1.29	2.40	8.36	3.30	84.63

FayCNS15						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1680	108	18	3	130	D	86.64	5.57	0.93	0.15	6.70
P		34	481	11	3	4	P	6.38	90.24	2.06	0.56	0.75
R		0	3	323	3	7	R	0.00	0.89	96.13	0.89	2.08
U		0	1	0	28	0	U	0.00	3.45	0.00	96.55	0.00
N		54	95	261	180	3738	N	1.25	2.20	6.03	4.16	86.37

FayCNS25						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1809	44	11	1	74	D	93.30	2.27	0.57	0.05	3.82
P		63	444	11	11	4	P	11.82	83.30	2.06	2.06	0.75
R		1	1	319	5	10	R	0.30	0.30	94.94	1.49	2.98
U		0	0	1	28	0	U	0.00	0.00	3.45	96.55	0.00
N		50	110	287	131	3750	N	1.16	2.54	6.63	3.03	86.65

FayC4.511						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1535	73	72	6	253	D	79.16	3.76	3.71	0.31	13.05
P		37	447	18	19	12	P	6.94	83.86	3.38	3.56	2.25
R		2	6	264	5	59	R	0.60	1.79	78.57	1.49	17.56



U	0	2	1	24	2	U	0.00	6.90	3.45	82.76	6.90
N	29	225	388	142	3544	N	0.67	5.20	8.96	3.28	81.89
<b>FayC4.515</b>						<b>Acierto</b>					
	D	P	R	U	N		D	P	R	U	N
D	1520	127	39	13	240	D	78.39	6.55	2.01	0.67	12.38
P	12	497	2	17	5	P	2.25	93.25	0.38	3.19	0.94
R	3	11	209	10	103	R	0.89	3.27	62.20	2.98	30.65
U	0	1	0	26	2	U	0.00	3.45	0.00	89.66	6.90
N	24	213	33	266	3792	N	0.55	4.92	0.76	6.15	87.62
<b>FayC4.525</b>						<b>Acierto</b>					
	D	P	R	U	N		D	P	R	U	N
D	1518	184	85	6	146	D	78.29	9.49	4.38	0.31	7.53
P	34	456	22	13	8	P	6.38	85.55	4.13	2.44	1.50
R	5	18	301	6	6	R	1.49	5.36	89.58	1.79	1.79
U	1	8	0	17	3	U	3.45	27.59	0.00	58.62	10.34
N	73	384	425	208	3238	N	1.69	8.87	9.82	4.81	74.82
<b>FayNB11</b>						<b>Acierto</b>					
	D	P	R	U	N		D	P	R	U	N
D	1602	120	36	39	142	D	82.62	6.19	1.86	2.01	7.32
P	40	404	13	70	6	P	7.50	75.80	2.44	13.13	1.13
R	0	14	272	38	12	R	0.00	4.17	80.95	11.31	3.57
U	0	0	6	23	0	U	0.00	0.00	20.69	79.31	0.00
N	57	198	347	72	3654	N	1.32	4.57	8.02	1.66	84.43
<b>FayNB15</b>						<b>Acierto</b>					
	D	P	R	U	N		D	P	R	U	N
D	1709	45	36	39	110	D	88.14	2.32	1.86	2.01	5.67
P	35	464	10	20	4	P	6.57	87.05	1.88	3.75	0.75
R	0	15	279	38	4	R	0.00	4.46	83.04	11.31	1.19
U	0	6	6	17	0	U	0.00	20.69	20.69	58.62	0.00
N	71	241	447	74	3495	N	1.64	5.57	10.33	1.71	80.75
<b>FayNB25</b>						<b>Acierto</b>					
	D	P	R	U	N		D	P	R	U	N
D	1709	14	59	39	118	D	88.14	0.72	3.04	2.01	6.09
P	31	446	35	16	5	P	5.82	83.68	6.57	3.00	0.94
R	0	8	285	38	5	R	0.00	2.38	84.82	11.31	1.49
U	0	6	6	17	0	U	0.00	20.69	20.69	58.62	0.00
N	71	133	542	63	3519	N	1.64	3.07	12.52	1.46	81.31
<b>MFrec11</b>						<b>Acierto</b>					
	D	P	R	U	N		D	P	R	U	N
D	3172	63	54	56	437	D	83.87	1.67	1.43	1.48	11.55
P	54	575	16	65	22	P	7.38	78.55	2.19	8.88	3.01
R	1	11	351	10	88	R	0.22	2.39	76.14	2.17	19.09
U	0	0	1	32	3	U	0.00	0.00	2.78	88.89	8.33
N	40	260	316	207	5150	N	0.67	4.35	5.29	3.47	86.22
<b>MFrec15</b>						<b>Acierto</b>					
	D	P	R	U	N		D	P	R	U	N
D	3219	64	121	53	325	D	85.11	1.69	3.20	1.40	8.59
P	19	596	29	86	2	P	2.60	81.42	3.96	11.75	0.27
R	0	1	452	4	4	R	0.00	0.22	98.05	0.87	0.87
U	0	0	2	32	2	U	0.00	0.00	5.56	88.89	5.56
N	31	277	783	284	4598	N	0.52	4.64	13.11	4.75	76.98
<b>MFrec25</b>						<b>Acierto</b>					
	D	P	R	U	N		D	P	R	U	N
D	3227	46	99	46	364	D	85.33	1.22	2.62	1.22	9.62
P	28	641	25	29	9	P	3.83	87.57	3.42	3.96	1.23
R	0	1	432	6	22	R	0.00	0.22	93.71	1.30	4.77
U	0	0	2	31	3	U	0.00	0.00	5.56	86.11	8.33
N	79	231	566	100	4997	N	1.32	3.87	9.48	1.67	83.66
<b>MFrecCFS11</b>						<b>Acierto</b>					
	D	P	R	U	N		D	P	R	U	N
D	3150	128	45	0	459	D	83.29	3.38	1.19	0.00	12.14
P	38	661	1	21	11	P	5.19	90.30	0.14	2.87	1.50
R	0	5	438	2	16	R	0.00	1.08	95.01	0.43	3.47
U	0	1	0	34	1	U	0.00	2.78	0.00	94.44	2.78
N	130	116	336	128	5263	N	2.18	1.94	5.63	2.14	88.11
<b>MFrecCFS15</b>						<b>Acierto</b>					
	D	P	R	U	N		D	P	R	U	N
D	3281	123	43	0	335	D	86.75	3.25	1.14	0.00	8.86
P	31	668	2	22	9	P	4.23	91.26	0.27	3.01	1.23

R	0	12	439	1	9	R	0.00	2.60	95.23	0.22	1.95
U	0	1	0	35	0	U	0.00	2.78	0.00	97.22	0.00
N	123	128	338	132	5252	N	2.06	2.14	5.66	2.21	87.93

<b>MFrecCFS25</b>						<b>Acierto</b>					
-------------------	--	--	--	--	--	----------------	--	--	--	--	--

	D	P	R	U	N		D	P	R	U	N
D	3450	58	43	0	231	D	91.22	1.53	1.14	0.00	6.11
P	27	680	1	22	2	P	3.69	92.90	0.14	3.01	0.27
R	0	2	447	5	7	R	0.00	0.43	96.96	1.08	1.52
U	0	0	0	36	0	U	0.00	0.00	0.00	100.00	0.00
N	159	115	362	138	5199	N	2.66	1.93	6.06	2.31	87.04

<b>MFrecCNS11</b>						<b>Acierto</b>					
-------------------	--	--	--	--	--	----------------	--	--	--	--	--

	D	P	R	U	N		D	P	R	U	N
D	3219	65	44	83	371	D	85.11	1.72	1.16	2.19	9.81
P	64	530	44	38	56	P	8.74	72.40	6.01	5.19	7.65
R	0	1	425	18	17	R	0.00	0.22	92.19	3.90	3.69
U	0	0	0	34	2	U	0.00	0.00	0.00	94.44	5.56
N	123	217	516	238	4879	N	2.06	3.63	8.64	3.98	81.68

<b>MFrecCNS15</b>						<b>Acierto</b>					
-------------------	--	--	--	--	--	----------------	--	--	--	--	--

	D	P	R	U	N		D	P	R	U	N
D	3149	93	47	62	431	D	83.26	2.46	1.24	1.64	11.40
P	51	601	16	41	23	P	6.97	82.10	2.19	5.60	3.14
R	0	2	393	25	41	R	0.00	0.43	85.25	5.42	8.89
U	0	0	1	35	0	U	0.00	0.00	2.78	97.22	0.00
N	107	181	357	311	5017	N	1.79	3.03	5.98	5.21	83.99

<b>MFrecCNS25</b>						<b>Acierto</b>					
-------------------	--	--	--	--	--	----------------	--	--	--	--	--

	D	P	R	U	N		D	P	R	U	N
D	3165	117	39	37	424	D	83.69	3.09	1.03	0.98	11.21
P	44	638	12	28	10	P	6.01	87.16	1.64	3.83	1.37
R	0	0	423	9	29	R	0.00	0.00	91.76	1.95	6.29
U	0	0	0	36	0	U	0.00	0.00	0.00	100.00	0.00
N	99	254	392	199	5029	N	1.66	4.25	6.56	3.33	84.20

<b>MFrecC4.511</b>						<b>Acierto</b>					
--------------------	--	--	--	--	--	----------------	--	--	--	--	--

	D	P	R	U	N		D	P	R	U	N
D	3000	160	99	57	466	D	79.32	4.23	2.62	1.51	12.32
P	24	618	22	63	5	P	3.28	84.43	3.01	8.61	0.68
R	0	8	339	20	94	R	0.00	1.74	73.54	4.34	20.39
U	0	0	1	34	1	U	0.00	0.00	2.78	94.44	2.78
N	21	288	338	202	5124	N	0.35	4.82	5.66	3.38	85.79

<b>MFrecC4.515</b>						<b>Acierto</b>					
--------------------	--	--	--	--	--	----------------	--	--	--	--	--

	D	P	R	U	N		D	P	R	U	N
D	3021	154	270	38	299	D	79.88	4.07	7.14	1.00	7.91
P	21	636	31	42	2	P	2.87	86.89	4.23	5.74	0.27
R	0	1	435	24	1	R	0.00	0.22	94.36	5.21	0.22
U	0	0	1	35	0	U	0.00	0.00	2.78	97.22	0.00
N	45	324	1253	241	4110	N	0.75	5.42	20.98	4.03	68.81

<b>MFrecC4.525</b>						<b>Acierto</b>					
--------------------	--	--	--	--	--	----------------	--	--	--	--	--

	D	P	R	U	N		D	P	R	U	N
D	3130	130	232	2	288	D	82.76	3.44	6.13	0.05	7.62
P	35	614	58	25	0	P	4.78	83.88	7.92	3.42	0.00
R	0	3	450	5	3	R	0.00	0.65	97.61	1.08	0.65
U	0	0	3	31	2	U	0.00	0.00	8.33	86.11	5.56
N	73	250	1041	134	4475	N	1.22	4.19	17.43	2.24	74.92

<b>MFrecNB11</b>						<b>Acierto</b>					
------------------	--	--	--	--	--	----------------	--	--	--	--	--

	D	P	R	U	N		D	P	R	U	N
D	3411	64	52	78	177	D	90.19	1.69	1.37	2.06	4.68
P	31	565	42	82	12	P	4.23	77.19	5.74	11.20	1.64
R	0	8	440	5	8	R	0.00	1.74	95.44	1.08	1.74
U	0	0	7	29	0	U	0.00	0.00	19.44	80.56	0.00
N	90	180	504	93	5106	N	1.51	3.01	8.44	1.56	85.48

<b>MFrecNB15</b>						<b>Acierto</b>					
------------------	--	--	--	--	--	----------------	--	--	--	--	--

	D	P	R	U	N		D	P	R	U	N
D	3229	236	57	77	183	D	85.38	6.24	1.51	2.04	4.84
P	23	605	15	81	8	P	3.14	82.65	2.05	11.07	1.09
R	0	21	428	5	7	R	0.00	4.56	92.84	1.08	1.52
U	0	0	7	29	0	U	0.00	0.00	19.44	80.56	0.00
N	92	310	480	90	5001	N	1.54	5.19	8.04	1.51	83.73

<b>MFrecNB25</b>						<b>Acierto</b>					
------------------	--	--	--	--	--	----------------	--	--	--	--	--

	D	P	R	U	N		D	P	R	U	N
D	3255	224	49	77	177	D	86.07	5.92	1.30	2.04	4.68

P	31	599	16	80	6	P	4.23	81.83	2.19	10.93	0.82
R	0	22	427	5	7	R	0.00	4.77	92.62	1.08	1.52
U	0	0	8	28	0	U	0.00	0.00	22.22	77.78	0.00
N	91	297	481	90	5014	N	1.52	4.97	8.05	1.51	83.94

Fay

Clonalg							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1718	2	58	0	161	D	88.60	0.10	2.99	0.00	8.30
P		166	214	3	0	150	P	31.14	40.15	0.56	0.00	28.14
R		20	4	189	0	123	R	5.95	1.19	56.25	0.00	36.61
U		8	7	5	0	9	U	27.59	24.14	17.24	0.00	31.03
N		166	124	156	0	3882	N	3.84	2.87	3.60	0.00	89.70

Genetico							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1561	14	0	0	364	D	80.51	0.72	0.00	0.00	18.77
P		127	186	12	1	207	P	23.83	34.90	2.25	0.19	38.84
R		0	1	1	0	334	R	0.00	0.30	0.30	0.00	99.40
U		0	8	0	3	18	U	0.00	27.59	0.00	10.34	62.07
N		17	130	4	5	4172	N	0.39	3.00	0.09	0.12	96.40

Furia							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1936	0	0	0	3	D	99.85	0.00	0.00	0.00	0.15
P		2	522	0	0	9	P	0.38	97.94	0.00	0.00	1.69
R		0	0	267	0	69	R	0.00	0.00	79.46	0.00	20.54
U		0	1	0	20	8	U	0.00	3.45	0.00	68.97	27.59
N		9	11	3	0	4305	N	0.21	0.25	0.07	0.00	99.47

Part							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1935	1	0	0	3	D	99.79	0.05	0.00	0.00	0.15
P		3	522	1	0	7	P	0.56	97.94	0.19	0.00	1.31
R		0	1	309	0	26	R	0.00	0.30	91.96	0.00	7.74
U		1	1	1	22	4	U	3.45	3.45	3.45	75.86	13.79
N		7	12	3	2	4304	N	0.16	0.28	0.07	0.05	99.45

Ripper							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1936	3	0	0	0	D	99.85	0.15	0.00	0.00	0.00
P		1	524	0	2	6	P	0.19	98.31	0.00	0.38	1.13
R		0	0	301	1	34	R	0.00	0.00	89.58	0.30	10.12
U		0	0	0	29	0	U	0.00	0.00	0.00	100.00	0.00
N		14	12	11	4	4287	N	0.32	0.28	0.25	0.09	99.05

RNDF							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1939	0	0	0	0	D	100.00	0.00	0.00	0.00	0.00
P		0	533	0	0	0	P	0.00	100.00	0.00	0.00	0.00
R		0	0	329	0	7	R	0.00	0.00	97.92	0.00	2.08
U		0	0	0	29	0	U	0.00	0.00	0.00	100.00	0.00
N		3	0	3	0	4322	N	0.07	0.00	0.07	0.00	99.86

C4.5							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1932	1	0	0	6	D	99.64	0.05	0.00	0.00	0.31
P		4	517	0	0	12	P	0.75	97.00	0.00	0.00	2.25
R		3	0	292	2	39	R	0.89	0.00	86.90	0.60	11.61
U		0	5	1	21	2	U	0.00	17.24	3.45	72.41	6.90
N		8	16	2	1	4301	N	0.18	0.37	0.05	0.02	99.38

NBTree							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1934	3	0	0	2	D	99.74	0.15	0.00	0.00	0.10
P		4	524	1	0	4	P	0.75	98.31	0.19	0.00	0.75
R		0	1	302	0	33	R	0.00	0.30	89.88	0.00	9.82
U		0	6	0	16	7	U	0.00	20.69	0.00	55.17	24.14
N		13	21	9	3	4282	N	0.30	0.49	0.21	0.07	98.94

CART							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1934	1	0	0	4	D	99.74	0.05	0.00	0.00	0.21
P		0	529	0	0	4	P	0.00	99.25	0.00	0.00	0.75
R		0	0	302	0	34	R	0.00	0.00	89.88	0.00	10.12
U		0	0	0	19	10	U	0.00	0.00	0.00	65.52	34.48
N		7	11	2	1	4307	N	0.16	0.25	0.05	0.02	99.51

SMOPoly							Acierto				
	D	P	R	U	N		D	P	R	U	N

D	1938	0	0	0	1	D	99.95	0.00	0.00	0.00	0.05
P	0	531	0	0	2	P	0.00	99.62	0.00	0.00	0.38
R	0	0	317	0	19	R	0.00	0.00	94.35	0.00	5.65
U	0	0	0	29	0	U	0.00	0.00	0.00	100.00	0.00
N	6	9	1	0	4312	N	0.14	0.21	0.02	0.00	99.63

**SMORBF** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1937	0	0	0	2	D	99.90	0.00	0.00	0.00	0.10
P	0	516	0	0	17	P	0.00	96.81	0.00	0.00	3.19
R	0	0	258	0	78	R	0.00	0.00	76.79	0.00	23.21
U	0	2	0	7	20	U	0.00	6.90	0.00	24.14	68.97
N	15	31	1	0	4281	N	0.35	0.72	0.02	0.00	98.91

**CSV-RBF** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1939	0	0	0	3	D	99.85	0.00	0.00	0.00	0.15
P	533	531	0	0	2	P	50.00	49.81	0.00	0.00	0.19
R	336	0	311	0	25	R	50.00	0.00	46.28	0.00	3.72
U	29	0	0	23	6	U	50.00	0.00	0.00	39.66	10.34
N	3692	9	2	0	4313	N	46.06	0.11	0.02	0.00	53.80

**CSV-SIGM** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	4	0	0	0	1935	D	0.21	0.00	0.00	0.00	99.79
P	0	0	0	0	533	P	0.00	0.00	0.00	0.00	100.00
R	0	0	0	0	336	R	0.00	0.00	0.00	0.00	100.00
U	0	0	0	0	29	U	0.00	0.00	0.00	0.00	100.00
N	0	0	0	0	4328	N	0.00	0.00	0.00	0.00	100.00

**PML** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1930	0	0	8	1	D	99.54	0.00	0.00	0.41	0.05
P	3	0	0	521	9	P	0.56	0.00	0.00	97.75	1.69
R	0	0	0	301	35	R	0.00	0.00	0.00	89.58	10.42
U	0	0	0	23	6	U	0.00	0.00	0.00	79.31	20.69
N	11	0	0	78	4239	N	0.25	0.00	0.00	1.80	97.94

**NAiveBayes** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1730	37	25	38	109	D	89.22	1.91	1.29	1.96	5.62
P	3	516	5	2	7	P	0.56	96.81	0.94	0.38	1.31
R	0	3	302	5	26	R	0.00	0.89	89.88	1.49	7.74
U	0	4	0	23	2	U	0.00	13.79	0.00	79.31	6.90
N	8	165	189	71	3895	N	0.18	3.81	4.37	1.64	90.00

**TAN** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1922	5	0	0	12	D	99.12	0.26	0.00	0.00	0.62
P	1	522	0	0	10	P	0.19	97.94	0.00	0.00	1.88
R	0	0	326	1	9	R	0.00	0.00	97.02	0.30	2.68
U	0	0	3	24	2	U	0.00	0.00	10.34	82.76	6.90
N	6	45	28	4	4245	N	0.14	1.04	0.65	0.09	98.08

**RBFNet** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1912	21	0	0	6	D	98.61	1.08	0.00	0.00	0.31
P	0	513	1	0	19	P	0.00	96.25	0.19	0.00	3.56
R	0	1	296	0	39	R	0.00	0.30	88.10	0.00	11.61
U	0	1	2	15	11	U	0.00	3.45	6.90	51.72	37.93
N	14	64	59	5	4186	N	0.32	1.48	1.36	0.12	96.72

**KNN-1** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1937	0	0	0	2	D	99.90	0.00	0.00	0.00	0.10
P	0	533	0	0	0	P	0.00	100.00	0.00	0.00	0.00
R	0	0	324	0	12	R	0.00	0.00	96.43	0.00	3.57
U	0	0	0	29	0	U	0.00	0.00	0.00	100.00	0.00
N	0	0	0	0	4328	N	0.00	0.00	0.00	0.00	100.00

**KNN-50** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1924	8	0	0	7	D	99.23	0.41	0.00	0.00	0.36
P	34	467	0	0	32	P	6.38	87.62	0.00	0.00	6.00
R	0	1	254	0	81	R	0.00	0.30	75.60	0.00	24.11

U	1	5	2	1	20	U	3.45	17.24	6.90	3.45	68.97
N	20	41	9	0	4258	N	0.46	0.95	0.21	0.00	98.38

## FayCFS

Clonalg						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1713	103	0	0	123	D	88.34	5.31	0.00	0.00	6.34
P		251	216	4	0	62	P	47.09	40.53	0.75	0.00	11.63
R		151	24	97	0	64	R	44.94	7.14	28.87	0.00	19.05
U		12	10	0	0	7	U	41.38	34.48	0.00	0.00	24.14
N		475	182	10	0	3661	N	10.98	4.21	0.23	0.00	84.59

Genetico						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1615	0	0	0	324	D	83.29	0.00	0.00	0.00	16.71
P		291	68	0	0	174	P	54.60	12.76	0.00	0.00	32.65
R		1	7	51	0	277	R	0.30	2.08	15.18	0.00	82.44
U		15	0	0	0	14	U	51.72	0.00	0.00	0.00	48.28
N		79	82	7	0	4160	N	1.83	1.89	0.16	0.00	96.12

Furia						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1935	3	0	0	1	D	99.79	0.15	0.00	0.00	0.05
P		13	499	0	0	21	P	2.44	93.62	0.00	0.00	3.94
R		1	0	256	0	79	R	0.30	0.00	76.19	0.00	23.51
U		0	5	0	15	9	U	0.00	17.24	0.00	51.72	31.03
N		20	28	1	3	4276	N	0.46	0.65	0.02	0.07	98.80

Part						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1936	1	0	0	2	D	99.85	0.05	0.00	0.00	0.10
P		2	520	1	0	10	P	0.38	97.56	0.19	0.00	1.88
R		1	1	290	0	44	R	0.30	0.30	86.31	0.00	13.10
U		1	5	0	17	6	U	3.45	17.24	0.00	58.62	20.69
N		16	26	8	4	4274	N	0.37	0.60	0.18	0.09	98.75

Ripper						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1929	9	0	0	1	D	99.48	0.46	0.00	0.00	0.05
P		3	487	0	2	41	P	0.56	91.37	0.00	0.38	7.69
R		0	0	259	0	77	R	0.00	0.00	77.08	0.00	22.92
U		0	0	0	23	6	U	0.00	0.00	0.00	79.31	20.69
N		16	26	27	6	4253	N	0.37	0.60	0.62	0.14	98.27

RNDF						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1938	1	0	0	0	D	99.95	0.05	0.00	0.00	0.00
P		1	527	0	0	5	P	0.19	98.87	0.00	0.00	0.94
R		0	0	307	0	29	R	0.00	0.00	91.37	0.00	8.63
U		0	0	1	24	4	U	0.00	0.00	3.45	82.76	13.79
N		13	6	9	3	4297	N	0.30	0.14	0.21	0.07	99.28

C4.5						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1933	3	0	0	3	D	99.69	0.15	0.00	0.00	0.15
P		6	512	0	0	15	P	1.13	96.06	0.00	0.00	2.81
R		0	0	299	0	37	R	0.00	0.00	88.99	0.00	11.01
U		0	6	1	18	4	U	0.00	20.69	3.45	62.07	13.79
N		16	14	7	4	4287	N	0.37	0.32	0.16	0.09	99.05

NBTree						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1933	4	0	0	2	D	99.69	0.21	0.00	0.00	0.10
P		8	509	1	0	15	P	1.50	95.50	0.19	0.00	2.81
R		0	0	294	0	42	R	0.00	0.00	87.50	0.00	12.50
U		1	6	0	7	15	U	3.45	20.69	0.00	24.14	51.72
N		18	24	15	2	4269	N	0.42	0.55	0.35	0.05	98.64

Cart						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		1935	1	0	0	3	D	99.79	0.05	0.00	0.00	0.15
P		3	520	0	0	10	P	0.56	97.56	0.00	0.00	1.88
R		1	0	302	1	32	R	0.30	0.00	89.88	0.30	9.52
U		0	0	1	21	7	U	0.00	0.00	3.45	72.41	24.14
N		12	13	10	3	4290	N	0.28	0.30	0.23	0.07	99.12

SMOPoly						Acierto					
	D	P	R	U	N		D	P	R	U	N

D	1934	1	0	0	4	D	99.74	0.05	0.00	0.00	0.21
P	3	514	0	0	16	P	0.56	96.44	0.00	0.00	3.00
R	0	0	301	0	35	R	0.00	0.00	89.58	0.00	10.42
U	0	0	0	23	6	U	0.00	0.00	0.00	79.31	20.69
N	13	21	15	3	4276	N	0.30	0.49	0.35	0.07	98.80

**SMORBF** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1928	6	0	0	5	D	99.43	0.31	0.00	0.00	0.26
P	13	501	0	0	19	P	2.44	94.00	0.00	0.00	3.56
R	0	0	244	0	92	R	0.00	0.00	72.62	0.00	27.38
U	0	15	0	0	14	U	0.00	51.72	0.00	0.00	48.28
N	20	52	4	0	4252	N	0.46	1.20	0.09	0.00	98.24

**CSV-RBF** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1938	1	0	0	0	D	99.95	0.05	0.00	0.00	0.00
P	2	523	0	0	8	P	0.38	98.12	0.00	0.00	1.50
R	0	1	298	1	36	R	0.00	0.30	88.69	0.30	10.71
U	0	0	1	22	6	U	0.00	0.00	3.45	75.86	20.69
N	18	19	19	4	4268	N	0.42	0.44	0.44	0.09	98.61

**CSV-SIGM** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	0	3	1	0	1935	D	0.00	0.15	0.05	0.00	99.79
P	0	0	0	0	533	P	0.00	0.00	0.00	0.00	100.00
R	0	0	0	0	336	R	0.00	0.00	0.00	0.00	100.00
U	0	0	0	0	29	U	0.00	0.00	0.00	0.00	100.00
N	0	0	0	0	4328	N	0.00	0.00	0.00	0.00	100.00

**PML** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1936	1	0	0	2	D	99.85	0.05	0.00	0.00	0.10
P	3	518	0	0	12	P	0.56	97.19	0.00	0.00	2.25
R	0	1	265	0	70	R	0.00	0.30	78.87	0.00	20.83
U	0	0	0	0	29	U	0.00	0.00	0.00	0.00	100.00
N	17	37	8	0	4266	N	0.39	0.85	0.18	0.00	98.57

**NiaveBayes** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1840	47	29	0	23	D	94.89	2.42	1.50	0.00	1.19
P	32	488	1	1	11	P	6.00	91.56	0.19	0.19	2.06
R	0	3	296	2	35	R	0.00	0.89	88.10	0.60	10.42
U	0	15	0	4	10	U	0.00	51.72	0.00	13.79	34.48
N	27	93	108	12	4088	N	0.62	2.15	2.50	0.28	94.45

**TAN** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1856	45	0	0	38	D	95.72	2.32	0.00	0.00	1.96
P	15	505	0	0	13	P	2.81	94.75	0.00	0.00	2.44
R	0	0	302	0	34	R	0.00	0.00	89.88	0.00	10.12
U	0	0	1	22	6	U	0.00	0.00	3.45	75.86	20.69
N	7	37	12	5	4267	N	0.16	0.85	0.28	0.12	98.59

**RBFNet** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1931	4	0	0	4	D	99.59	0.21	0.00	0.00	0.21
P	6	506	0	0	21	P	1.13	94.93	0.00	0.00	3.94
R	0	0	281	0	55	R	0.00	0.00	83.63	0.00	16.37
U	0	0	0	20	9	U	0.00	0.00	0.00	68.97	31.03
N	24	52	53	3	4196	N	0.55	1.20	1.22	0.07	96.95

**KNN-1** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1938	1	0	0	0	D	99.95	0.05	0.00	0.00	0.00
P	1	527	0	0	5	P	0.19	98.87	0.00	0.00	0.94
R	0	0	306	1	29	R	0.00	0.00	91.07	0.30	8.63
U	0	0	0	25	4	U	0.00	0.00	0.00	86.21	13.79
N	13	6	8	3	4298	N	0.30	0.14	0.18	0.07	99.31

**KNN-50** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1931	1	0	0	7	D	99.59	0.05	0.00	0.00	0.36
P	42	457	0	0	34	P	7.88	85.74	0.00	0.00	6.38
R	11	1	244	0	80	R	3.27	0.30	72.62	0.00	23.81



U	0	6	0	0	23	U	0.00	20.69	0.00	0.00	79.31
N	30	65	18	0	4215	N	0.69	1.50	0.42	0.00	97.39

FayCNS

Clonalg							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1521	105	32	0	281	D	78.44	5.42	1.65	0.00	14.49
P		80	266	19	0	168	P	15.01	49.91	3.56	0.00	31.52
R		5	24	44	0	263	R	1.49	7.14	13.10	0.00	78.27
U		5	9	1	0	14	U	17.24	31.03	3.45	0.00	48.28
N		36	123	164	0	4005	N	0.83	2.84	3.79	0.00	92.54

Genetico							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1586	17	77	1	258	D	81.79	0.88	3.97	0.05	13.31
P		52	337	6	15	123	P	9.76	63.23	1.13	2.81	23.08
R		3	5	97	2	229	R	0.89	1.49	28.87	0.60	68.15
U		2	6	1	9	11	U	6.90	20.69	3.45	31.03	37.93
N		64	71	18	4	4171	N	1.48	1.64	0.42	0.09	96.37

Furia							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1927	2	0	0	10	D	99.38	0.10	0.00	0.00	0.52
P		4	522	1	0	6	P	0.75	97.94	0.19	0.00	1.13
R		0	0	244	0	92	R	0.00	0.00	72.62	0.00	27.38
U		0	3	0	16	10	U	0.00	10.34	0.00	55.17	34.48
N		9	14	0	0	4305	N	0.21	0.32	0.00	0.00	99.47

Part							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1931	3	0	0	5	D	99.59	0.15	0.00	0.00	0.26
P		1	524	2	0	6	P	0.19	98.31	0.38	0.00	1.13
R		1	1	302	0	32	R	0.30	0.30	89.88	0.00	9.52
U		1	1	2	18	7	U	3.45	3.45	6.90	62.07	24.14
N		13	21	6	1	4287	N	0.30	0.49	0.14	0.02	99.05

Ripper							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1930	4	1	0	4	D	99.54	0.21	0.05	0.00	0.21
P		1	524	1	1	6	P	0.19	98.31	0.19	0.19	1.13
R		0	1	293	0	42	R	0.00	0.30	87.20	0.00	12.50
U		0	4	1	16	8	U	0.00	13.79	3.45	55.17	27.59
N		13	18	19	5	4273	N	0.30	0.42	0.44	0.12	98.73

RNDF							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1939	0	0	0	0	D	100.00	0.00	0.00	0.00	0.00
P		0	533	0	0	0	P	0.00	100.00	0.00	0.00	0.00
R		0	0	329	0	7	R	0.00	0.00	97.92	0.00	2.08
U		0	0	0	29	0	U	0.00	0.00	0.00	100.00	0.00
N		3	0	3	0	4322	N	0.07	0.00	0.07	0.00	99.86

C4.5							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1924	2	0	0	13	D	99.23	0.10	0.00	0.00	0.67
P		3	520	2	0	8	P	0.56	97.56	0.38	0.00	1.50
R		0	0	296	0	40	R	0.00	0.00	88.10	0.00	11.90
U		0	6	0	18	5	U	0.00	20.69	0.00	62.07	17.24
N		12	22	2	0	4292	N	0.28	0.51	0.05	0.00	99.17

NBTree							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1935	1	0	0	3	D	99.79	0.05	0.00	0.00	0.15
P		2	525	1	0	5	P	0.38	98.50	0.19	0.00	0.94
R		0	1	303	0	32	R	0.00	0.30	90.18	0.00	9.52
U		1	7	0	15	6	U	3.45	24.14	0.00	51.72	20.69
N		10	21	7	1	4289	N	0.23	0.49	0.16	0.02	99.10

CART							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1931	1	1	0	6	D	99.59	0.05	0.05	0.00	0.31
P		1	527	0	0	5	P	0.19	98.87	0.00	0.00	0.94
R		1	1	295	1	38	R	0.30	0.30	87.80	0.30	11.31
U		0	1	0	16	12	U	0.00	3.45	0.00	55.17	41.38
N		8	7	2	2	4309	N	0.18	0.16	0.05	0.05	99.56

SMOPoly							Acierto					
	D	P	R	U	N		D	P	R	U	N	

D	1937	0	0	0	2	D	99.90	0.00	0.00	0.00	0.10
P	0	531	0	0	2	P	0.00	99.62	0.00	0.00	0.38
R	0	1	313	0	22	R	0.00	0.30	93.15	0.00	6.55
U	0	0	0	24	5	U	0.00	0.00	0.00	82.76	17.24
N	9	14	1	0	4304	N	0.21	0.32	0.02	0.00	99.45

**SMORBF** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1926	4	0	0	9	D	99.33	0.21	0.00	0.00	0.46
P	1	520	0	0	12	P	0.19	97.56	0.00	0.00	2.25
R	0	1	249	0	86	R	0.00	0.30	74.11	0.00	25.60
U	0	8	0	0	21	U	0.00	27.59	0.00	0.00	72.41
N	15	44	1	0	4268	N	0.35	1.02	0.02	0.00	98.61

**CSV-RBF** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1936	0	0	1561	3	D	55.31	0.00	0.00	44.60	0.09
P	0	532	0	485	1	P	0.00	52.26	0.00	47.64	0.10
R	0	0	316	317	20	R	0.00	0.00	48.39	48.55	3.06
U	0	0	0	28	5	U	0.00	0.00	0.00	84.85	15.15
N	4	5	3	2720	4316	N	0.06	0.07	0.04	38.59	61.24

**CSV-SIGM** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	0	0	0	0	1939	D	0.00	0.00	0.00	0.00	100.00
P	0	0	0	0	533	P	0.00	0.00	0.00	0.00	100.00
R	0	0	0	0	336	R	0.00	0.00	0.00	0.00	100.00
U	0	0	0	0	29	U	0.00	0.00	0.00	0.00	100.00
N	0	0	0	0	4328	N	0.00	0.00	0.00	0.00	100.00

**PML** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1935	0	1	0	3	D	99.79	0.00	0.05	0.00	0.15
P	7	513	0	0	13	P	1.31	96.25	0.00	0.00	2.44
R	157	1	46	0	132	R	46.73	0.30	13.69	0.00	39.29
U	1	0	0	0	28	U	3.45	0.00	0.00	0.00	96.55
N	29	36	0	0	4263	N	0.67	0.83	0.00	0.00	98.50

**NaiveBayes** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1688	37	11	46	157	D	87.06	1.91	0.57	2.37	8.10
P	1	523	3	3	3	P	0.19	98.12	0.56	0.56	0.56
R	0	3	296	5	32	R	0.00	0.89	88.10	1.49	9.52
U	0	6	1	19	3	U	0.00	20.69	3.45	65.52	10.34
N	14	131	138	173	3872	N	0.32	3.03	3.19	4.00	89.46

**TAN** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1911	5	0	0	23	D	98.56	0.26	0.00	0.00	1.19
P	1	523	0	0	9	P	0.19	98.12	0.00	0.00	1.69
R	0	0	323	1	12	R	0.00	0.00	96.13	0.30	3.57
U	0	2	3	19	5	U	0.00	6.90	10.34	65.52	17.24
N	12	46	24	1	4245	N	0.28	1.06	0.55	0.02	98.08

**RBFNet** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1881	33	0	0	25	D	97.01	1.70	0.00	0.00	1.29
P	2	513	0	0	18	P	0.38	96.25	0.00	0.00	3.38
R	0	1	310	0	25	R	0.00	0.30	92.26	0.00	7.44
U	0	5	3	11	10	U	0.00	17.24	10.34	37.93	34.48
N	17	70	39	5	4197	N	0.39	1.62	0.90	0.12	96.97

**KNN-1** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1937	0	0	0	2	D	99.90	0.00	0.00	0.00	0.10
P	0	533	0	0	0	P	0.00	100.00	0.00	0.00	0.00
R	0	0	324	0	12	R	0.00	0.00	96.43	0.00	3.57
U	0	0	0	29	0	U	0.00	0.00	0.00	100.00	0.00
N	0	0	1	0	4327	N	0.00	0.00	0.02	0.00	99.98

**KNN-50** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1883	12	0	0	44	D	97.11	0.62	0.00	0.00	2.27
P	18	490	0	0	25	P	3.38	91.93	0.00	0.00	4.69
R	0	2	239	0	95	R	0.00	0.60	71.13	0.00	28.27

U	0	7	1	0	21	U	0.00	24.14	3.45	0.00	72.41
N	28	29	10	0	4261	N	0.65	0.67	0.23	0.00	98.45

## FayC4.5

Clonalg							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1674	7	47	0	211	D	86.33	0.36	2.42	0.00	10.88
P		163	217	5	0	148	P	30.58	40.71	0.94	0.00	27.77
R		15	4	167	0	150	R	4.46	1.19	49.70	0.00	44.64
U		10	5	4	0	10	U	34.48	17.24	13.79	0.00	34.48
N		150	81	128	0	3969	N	3.47	1.87	2.96	0.00	91.71

Genetico							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1591	8	35	18	287	D	82.05	0.41	1.81	0.93	14.80
P		125	111	0	14	283	P	23.45	20.83	0.00	2.63	53.10
R		1	1	28	1	305	R	0.30	0.30	8.33	0.30	90.77
U		0	0	0	2	27	U	0.00	0.00	0.00	6.90	93.10
N		190	25	19	2	4092	N	4.39	0.58	0.44	0.05	94.55

Furia							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1929	0	0	0	10	D	99.48	0.00	0.00	0.00	0.52
P		3	509	0	0	21	P	0.56	95.50	0.00	0.00	3.94
R		0	0	261	0	75	R	0.00	0.00	77.68	0.00	22.32
U		0	0	0	21	8	U	0.00	0.00	0.00	72.41	27.59
N		6	17	6	1	4298	N	0.14	0.39	0.14	0.02	99.31

Part							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1930	1	0	0	8	D	99.54	0.05	0.00	0.00	0.41
P		3	514	0	0	16	P	0.56	96.44	0.00	0.00	3.00
R		1	2	298	1	34	R	0.30	0.60	88.69	0.30	10.12
U		0	0	1	20	8	U	0.00	0.00	3.45	68.97	27.59
N		11	22	17	1	4277	N	0.25	0.51	0.39	0.02	98.82

Ripper							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1935	2	0	0	2	D	99.79	0.10	0.00	0.00	0.10
P		0	520	0	0	13	P	0.00	97.56	0.00	0.00	2.44
R		0	0	306	1	29	R	0.00	0.00	91.07	0.30	8.63
U		0	0	0	26	3	U	0.00	0.00	0.00	89.66	10.34
N		14	23	32	3	4256	N	0.32	0.53	0.74	0.07	98.34

RNDF							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1938	0	0	0	1	D	99.95	0.00	0.00	0.00	0.05
P		0	533	0	0	0	P	0.00	100.00	0.00	0.00	0.00
R		0	0	327	0	9	R	0.00	0.00	97.32	0.00	2.68
U		0	0	0	29	0	U	0.00	0.00	0.00	100.00	0.00
N		3	2	8	0	4315	N	0.07	0.05	0.18	0.00	99.70

C4.5							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1933	1	0	0	5	D	99.69	0.05	0.00	0.00	0.26
P		5	509	0	0	19	P	0.94	95.50	0.00	0.00	3.56
R		1	0	294	2	39	R	0.30	0.00	87.50	0.60	11.61
U		0	5	3	18	3	U	0.00	17.24	10.34	62.07	10.34
N		16	16	15	1	4280	N	0.37	0.37	0.35	0.02	98.89

NBTree							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1936	0	0	0	3	D	99.85	0.00	0.00	0.00	0.15
P		15	503	0	0	15	P	2.81	94.37	0.00	0.00	2.81
R		0	0	298	2	36	R	0.00	0.00	88.69	0.60	10.71
U		0	0	2	23	4	U	0.00	0.00	6.90	79.31	13.79
N		16	11	14	1	4286	N	0.37	0.25	0.32	0.02	99.03

CART							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1929	4	0	0	6	D	99.48	0.21	0.00	0.00	0.31
P		1	525	0	0	7	P	0.19	98.50	0.00	0.00	1.31
R		0	2	307	1	26	R	0.00	0.60	91.37	0.30	7.74
U		0	0	0	25	4	U	0.00	0.00	0.00	86.21	13.79
N		16	5	13	4	4290	N	0.37	0.12	0.30	0.09	99.12

SMOPoly							Acierto					
	D	P	R	U	N		D	P	R	U	N	

D	1935	0	0	0	4	D	99.79	0.00	0.00	0.00	0.21
P	0	517	0	0	16	P	0.00	97.00	0.00	0.00	3.00
R	0	0	310	0	26	R	0.00	0.00	92.26	0.00	7.74
U	0	0	0	27	2	U	0.00	0.00	0.00	93.10	6.90
N	11	18	24	2	4273	N	0.25	0.42	0.55	0.05	98.73

**SMORBF** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1935	1	0	0	3	D	99.79	0.05	0.00	0.00	0.15
P	3	502	0	0	28	P	0.56	94.18	0.00	0.00	5.25
R	1	1	241	0	93	R	0.30	0.30	71.73	0.00	27.68
U	0	8	0	0	21	U	0.00	27.59	0.00	0.00	72.41
N	18	29	5	0	4276	N	0.42	0.67	0.12	0.00	98.80

**CSV-RBF** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1931	5	0	0	3	D	99.59	0.26	0.00	0.00	0.15
P	0	526	0	0	7	P	0.00	98.69	0.00	0.00	1.31
R	0	0	307	0	29	R	0.00	0.00	91.37	0.00	8.63
U	0	0	0	22	7	U	0.00	0.00	0.00	75.86	24.14
N	6	10	11	0	4301	N	0.14	0.23	0.25	0.00	99.38

**CSV-SIGM** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	0	0	0	0	1939	D	0.00	0.00	0.00	0.00	100.00
P	0	0	0	0	533	P	0.00	0.00	0.00	0.00	100.00
R	0	0	0	0	336	R	0.00	0.00	0.00	0.00	100.00
U	0	0	0	0	29	U	0.00	0.00	0.00	0.00	100.00
N	0	0	0	0	4328	N	0.00	0.00	0.00	0.00	100.00

**PML** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1934	2	0	1	2	D	99.74	0.10	0.00	0.05	0.10
P	3	490	1	3	36	P	0.56	91.93	0.19	0.56	6.75
R	0	0	268	15	53	R	0.00	0.00	79.76	4.46	15.77
U	0	1	3	14	11	U	0.00	3.45	10.34	48.28	37.93
N	16	23	12	5	4272	N	0.37	0.53	0.28	0.12	98.71

**NaiveBayes** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1600	98	24	26	191	D	82.52	5.05	1.24	1.34	9.85
P	5	513	8	3	4	P	0.94	96.25	1.50	0.56	0.75
R	0	4	299	3	30	R	0.00	1.19	88.99	0.89	8.93
U	0	8	3	16	2	U	0.00	27.59	10.34	55.17	6.90
N	23	200	134	74	3897	N	0.53	4.62	3.10	1.71	90.04

**TAN** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1913	4	0	0	22	D	98.66	0.21	0.00	0.00	1.13
P	2	518	1	0	12	P	0.38	97.19	0.19	0.00	2.25
R	0	1	313	1	21	R	0.00	0.30	93.15	0.30	6.25
U	0	1	3	21	4	U	0.00	3.45	10.34	72.41	13.79
N	22	45	43	3	4215	N	0.51	1.04	0.99	0.07	97.39

**RBFNet** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1901	17	0	0	21	D	98.04	0.88	0.00	0.00	1.08
P	0	507	0	1	25	P	0.00	95.12	0.00	0.19	4.69
R	0	1	301	1	33	R	0.00	0.30	89.58	0.30	9.82
U	0	6	2	15	6	U	0.00	20.69	6.90	51.72	20.69
N	19	46	47	6	4210	N	0.44	1.06	1.09	0.14	97.27

**KNN-1** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1938	0	0	0	1	D	99.95	0.00	0.00	0.00	0.05
P	0	532	0	0	1	P	0.00	99.81	0.00	0.00	0.19
R	0	0	320	0	16	R	0.00	0.00	95.24	0.00	4.76
U	0	0	0	29	0	U	0.00	0.00	0.00	100.00	0.00
N	3	0	3	0	4322	N	0.07	0.00	0.07	0.00	99.86

**KNN-50** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	1890	11	0	0	38	D	97.47	0.57	0.00	0.00	1.96
P	20	471	0	0	42	P	3.75	88.37	0.00	0.00	7.88
R	0	1	235	0	100	R	0.00	0.30	69.94	0.00	29.76

U	4	7	2	0	16	U	13.79	24.14	6.90	0.00	55.17
N	20	40	12	0	4256	N	0.46	0.92	0.28	0.00	98.34

FayNB

Clonalg							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1657	12	7	0	263	D	85.46	0.62	0.36	0.00	13.56
P		244	40	28	0	221	P	45.78	7.50	5.25	0.00	41.46
R		4	1	136	0	195	R	1.19	0.30	40.48	0.00	58.04
U		8	0	2	0	19	U	27.59	0.00	6.90	0.00	65.52
N		83	7	102	0	4136	N	1.92	0.16	2.36	0.00	95.56

Genetico							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1563	11	12	0	353	D	80.61	0.57	0.62	0.00	18.21
P		93	205	0	0	235	P	17.45	38.46	0.00	0.00	44.09
R		2	9	51	0	274	R	0.60	2.68	15.18	0.00	81.55
U		0	6	0	3	20	U	0.00	20.69	0.00	10.34	68.97
N		23	177	10	0	4118	N	0.53	4.09	0.23	0.00	95.15

Furia							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1821	0	0	0	118	D	93.91	0.00	0.00	0.00	6.09
P		39	341	2	0	151	P	7.32	63.98	0.38	0.00	28.33
R		0	1	186	1	148	R	0.00	0.30	55.36	0.30	44.05
U		1	5	0	15	8	U	3.45	17.24	0.00	51.72	27.59
N		39	33	17	0	4239	N	0.90	0.76	0.39	0.00	97.94

Part							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1828	5	0	0	106	D	94.28	0.26	0.00	0.00	5.47
P		15	478	2	0	38	P	2.81	89.68	0.38	0.00	7.13
R		1	2	225	2	106	R	0.30	0.60	66.96	0.60	31.55
U		2	5	2	15	5	U	6.90	17.24	6.90	51.72	17.24
N		38	41	34	2	4213	N	0.88	0.95	0.79	0.05	97.34

Ripper							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1810	0	0	0	129	D	93.35	0.00	0.00	0.00	6.65
P		7	357	2	0	167	P	1.31	66.98	0.38	0.00	31.33
R		0	1	226	1	108	R	0.00	0.30	67.26	0.30	32.14
U		1	5	2	16	5	U	3.45	17.24	6.90	55.17	17.24
N		35	52	35	0	4206	N	0.81	1.20	0.81	0.00	97.18

RNDF							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1828	5	0	0	106	D	94.28	0.26	0.00	0.00	5.47
P		9	491	2	0	31	P	1.69	92.12	0.38	0.00	5.82
R		0	1	230	1	104	R	0.00	0.30	68.45	0.30	30.95
U		1	5	2	18	3	U	3.45	17.24	6.90	62.07	10.34
N		33	36	35	1	4223	N	0.76	0.83	0.81	0.02	97.57

C4.5							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1824	7	0	0	108	D	94.07	0.36	0.00	0.00	5.57
P		9	487	2	0	35	P	1.69	91.37	0.38	0.00	6.57
R		1	1	225	1	107	R	0.30	0.30	67.16	0.30	31.94
U		2	6	2	16	4	U	6.67	20.00	6.67	53.33	13.33
N		36	39	33	0	4217	N	0.83	0.90	0.76	0.00	97.50

NBT							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1824	8	0	0	107	D	94.07	0.41	0.00	0.00	5.52
P		37	438	2	0	56	P	6.94	82.18	0.38	0.00	10.51
R		0	0	227	2	107	R	0.00	0.00	67.56	0.60	31.85
U		1	6	2	15	5	U	3.45	20.69	6.90	51.72	17.24
N		37	27	34	4	4226	N	0.85	0.62	0.79	0.09	97.64

CART							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		1827	6	0	0	106	D	94.22	0.31	0.00	0.00	5.47
P		11	487	2	0	33	P	2.06	91.37	0.38	0.00	6.19
R		1	2	226	1	106	R	0.30	0.60	67.26	0.30	31.55
U		2	5	2	13	7	U	6.90	17.24	6.90	44.83	24.14
N		37	39	35	0	4217	N	0.85	0.90	0.81	0.00	97.44

SMOPoly							Acierto					
	D	P	R	U	N		D	P	R	U	N	



D	1788	4	0	0	147	D	92.21	0.21	0.00	0.00	7.58
P	11	480	1	0	41	P	2.06	90.06	0.19	0.00	7.69
R	0	1	256	1	78	R	0.00	0.30	76.19	0.30	23.21
U	0	5	2	13	9	U	0.00	17.24	6.90	44.83	31.03
N	27	54	64	2	4181	N	0.62	1.25	1.48	0.05	96.60

SMORBF						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	1753	6	0	0	180	D	90.41	0.31	0.00	0.00	9.28
P	26	410	0	0	97	P	4.88	76.92	0.00	0.00	18.20
R	1	5	123	0	207	R	0.30	1.49	36.61	0.00	61.61
U	0	15	0	0	14	U	0.00	51.72	0.00	0.00	48.28
N	25	60	9	0	4234	N	0.58	1.39	0.21	0.00	97.83

CSV-RBF						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	1825	7	0	0	107	D	94.12	0.36	0.00	0.00	5.52
P	14	464	2	0	53	P	2.63	87.05	0.38	0.00	9.94
R	0	1	225	0	110	R	0.00	0.30	66.96	0.00	32.74
U	2	6	2	10	9	U	6.90	20.69	6.90	34.48	31.03
N	37	25	38	2	4226	N	0.85	0.58	0.88	0.05	97.64

CSV-SIGM						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	1805	0	0	0	1916	D	48.51	0.00	0.00	0.00	51.49
P	518	0	0	0	533	P	49.29	0.00	0.00	0.00	50.71
R	328	1	0	0	335	R	49.40	0.15	0.00	0.00	50.45
U	28	0	0	0	29	U	49.12	0.00	0.00	0.00	50.88
N	3122	1	0	0	4325	N	41.92	0.01	0.00	0.00	58.07

PML						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	1832	2	0	0	105	D	94.48	0.10	0.00	0.00	5.42
P	12	484	2	0	35	P	2.25	90.81	0.38	0.00	6.57
R	1	1	234	1	99	R	0.30	0.30	69.64	0.30	29.46
U	1	5	2	14	7	U	3.45	17.24	6.90	48.28	24.14
N	37	41	40	0	4210	N	0.85	0.95	0.92	0.00	97.27

NaiveBayes						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	1645	5	0	0	289	D	84.84	0.26	0.00	0.00	14.90
P	46	456	1	0	30	P	8.63	85.55	0.19	0.00	5.63
R	0	9	195	1	131	R	0.00	2.68	58.04	0.30	38.99
U	0	15	0	4	10	U	0.00	51.72	0.00	13.79	34.48
N	12	141	48	0	4127	N	0.28	3.26	1.11	0.00	95.36

TAN						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	1815	12	0	0	112	D	93.60	0.62	0.00	0.00	5.78
P	33	435	1	0	64	P	6.19	81.61	0.19	0.00	12.01
R	0	2	226	1	107	R	0.00	0.60	67.26	0.30	31.85
U	1	6	2	13	7	U	3.45	20.69	6.90	44.83	24.14
N	32	22	36	2	4236	N	0.74	0.51	0.83	0.05	97.87

RBFNet						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	1812	20	0	0	107	D	93.45	1.03	0.00	0.00	5.52
P	22	462	3	0	46	P	4.13	86.68	0.56	0.00	8.63
R	0	4	245	1	86	R	0.00	1.19	72.92	0.30	25.60
U	1	6	2	14	6	U	3.45	20.69	6.90	48.28	20.69
N	36	74	67	0	4151	N	0.83	1.71	1.55	0.00	95.91

KNN-1						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	1826	5	0	0	108	D	94.17	0.26	0.00	0.00	5.57
P	8	490	2	0	33	P	1.50	91.93	0.38	0.00	6.19
R	0	1	230	1	104	R	0.00	0.30	68.45	0.30	30.95
U	1	5	2	17	4	U	3.45	17.24	6.90	58.62	13.79
N	31	36	34	0	4227	N	0.72	0.83	0.79	0.00	97.67

KNN-50						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	1803	10	0	0	126	D	92.99	0.52	0.00	0.00	6.50
P	60	364	3	0	106	P	11.26	68.29	0.56	0.00	19.89
R	2	1	176	0	157	R	0.60	0.30	52.38	0.00	46.73

U	2	11	0	0	16	U	6.90	37.93	0.00	0.00	55.17
N	52	47	40	0	4189	N	1.20	1.09	0.92	0.00	96.79

Frec

Clonalg							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		3096	0	9	0	677	D	81.86	0.00	0.24	0.00	17.90
P		254	50	22	0	406	P	34.70	6.83	3.01	0.00	55.46
R		2	0	68	0	391	R	0.43	0.00	14.75	0.00	84.82
U		8	0	0	0	28	U	22.22	0.00	0.00	0.00	77.78
N		14	0	23	0	5936	N	0.23	0.00	0.39	0.00	99.38

Genetico							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		3088	16	0	2	676	D	81.65	0.42	0.00	0.05	17.87
P		312	160	0	17	243	P	42.62	21.86	0.00	2.32	33.20
R		0	3	112	8	338	R	0.00	0.65	24.30	1.74	73.32
U		1	0	0	24	11	U	2.78	0.00	0.00	66.67	30.56
N		123	184	50	25	5591	N	2.06	3.08	0.84	0.42	93.60

Furia							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		3779	0	0	2342	3	D	61.71	0.00	0.00	38.24	0.05
P		2	722	0	621	8	P	0.15	53.36	0.00	45.90	0.59
R		0	0	353	392	108	R	0.00	0.00	41.38	45.96	12.66
U		0	2	0	35	7	U	0.00	4.55	0.00	79.55	15.91
N		9	17	3	3279	5944	N	0.10	0.18	0.03	35.44	64.25

Part							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		3776	1	0	0	5	D	99.84	0.03	0.00	0.00	0.13
P		12	707	0	0	13	P	1.64	96.58	0.00	0.00	1.78
R		3	0	425	0	33	R	0.65	0.00	92.19	0.00	7.16
U		0	0	1	27	8	U	0.00	0.00	2.78	75.00	22.22
N		17	20	13	0	5923	N	0.28	0.33	0.22	0.00	99.16

Ripper							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		3780	0	0	0	2	D	99.95	0.00	0.00	0.00	0.05
P		3	719	1	0	9	P	0.41	98.22	0.14	0.00	1.23
R		0	0	431	1	29	R	0.00	0.00	93.49	0.22	6.29
U		0	0	1	32	3	U	0.00	0.00	2.78	88.89	8.33
N		16	15	60	4	5878	N	0.27	0.25	1.00	0.07	98.41

RNDF							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		3775	4	0	0	3	D	99.81	0.11	0.00	0.00	0.08
P		4	725	0	0	3	P	0.55	99.04	0.00	0.00	0.41
R		1	1	413	0	46	R	0.22	0.22	89.59	0.00	9.98
U		0	6	0	21	9	U	0.00	16.67	0.00	58.33	25.00
N		15	14	13	1	5930	N	0.25	0.23	0.22	0.02	99.28

C4.5							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		3775	3	0	0	4	D	99.81	0.08	0.00	0.00	0.11
P		7	707	0	0	18	P	0.96	96.58	0.00	0.00	2.46
R		1	0	420	1	39	R	0.22	0.00	91.11	0.22	8.46
U		0	6	0	12	18	U	0.00	16.67	0.00	33.33	50.00
N		18	49	7	4	5895	N	0.30	0.82	0.12	0.07	98.69

NBTree							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		3775	4	0	0	3	D	99.81	0.11	0.00	0.00	0.08
P		4	725	0	0	3	P	0.55	99.04	0.00	0.00	0.41
R		1	1	413	0	46	R	0.22	0.22	89.59	0.00	9.98
U		0	6	0	21	9	U	0.00	16.67	0.00	58.33	25.00
N		15	14	13	1	5930	N	0.25	0.23	0.22	0.02	99.28

CART							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		3779	0	0	0	3	D	99.92	0.00	0.00	0.00	0.08
P		4	726	0	0	2	P	0.55	99.18	0.00	0.00	0.27
R		0	0	430	1	30	R	0.00	0.00	93.28	0.22	6.51
U		1	2	0	29	4	U	2.78	5.56	0.00	80.56	11.11
N		10	7	8	2	5946	N	0.17	0.12	0.13	0.03	99.55

SMOPoly							Acierto					
---------	--	--	--	--	--	--	---------	--	--	--	--	--

	D	P	R	U	N		D	P	R	U	N	
D	3781	0	0	0	0	1	D	99.97	0.00	0.00	0.00	0.03
P	0	731	0	0	0	1	P	0.00	99.86	0.00	0.00	0.14
R	0	0	435	0	0	26	R	0.00	0.00	94.36	0.00	5.64
U	0	0	0	36	0	0	U	0.00	0.00	0.00	100.00	0.00
N	6	8	3	0	5956		N	0.10	0.13	0.05	0.00	99.72

**SMORBF** Acierto

	D	P	R	U	N		D	P	R	U	N	
D	3776	0	0	0	0	6	D	99.84	0.00	0.00	0.00	0.16
P	0	712	0	0	0	20	P	0.00	97.27	0.00	0.00	2.73
R	0	0	390	0	0	71	R	0.00	0.00	84.60	0.00	15.40
U	0	1	1	20	14		U	0.00	2.78	2.78	55.56	38.89
N	24	41	3	0	5905		N	0.40	0.69	0.05	0.00	98.86

**CSV-RBF** Acierto

	D	P	R	U	N		D	P	R	U	N	
D	3780	0	0	0	0	2	D	99.95	0.00	0.00	0.00	0.05
P	0	731	0	0	0	1	P	0.00	99.86	0.00	0.00	0.14
R	0	0	431	0	0	30	R	0.00	0.00	93.49	0.00	6.51
U	0	0	0	31	5		U	0.00	0.00	0.00	86.11	13.89
N	4	3	3	0	5963		N	0.07	0.05	0.05	0.00	99.83

**CSV-SIGM** Acierto

	D	P	R	U	N		D	P	R	U	N	
D	0	0	0	0	0	3782	D	0.00	0.00	0.00	0.00	100.00
P	0	0	0	0	0	732	P	0.00	0.00	0.00	0.00	100.00
R	0	0	0	0	0	461	R	0.00	0.00	0.00	0.00	100.00
U	0	0	0	0	0	36	U	0.00	0.00	0.00	0.00	100.00
N	0	0	0	0	0	5973	N	0.00	0.00	0.00	0.00	100.00

**PML** Acierto

	D	P	R	U	N		D	P	R	U	N	
D	0	0	0	0	0	3782	D	0.00	0.00	0.00	0.00	100.00
P	0	0	0	0	0	732	P	0.00	0.00	0.00	0.00	100.00
R	0	0	0	0	0	461	R	0.00	0.00	0.00	0.00	100.00
U	0	0	0	0	0	36	U	0.00	0.00	0.00	0.00	100.00
N	0	0	0	0	0	5973	N	0.00	0.00	0.00	0.00	100.00

**NaiveBayes** Acierto

	D	P	R	U	N		D	P	R	U	N	
D	3251	97	55	49	330		D	85.96	2.56	1.45	1.30	8.73
P	3	717	7	1	4		P	0.41	97.95	0.96	0.14	0.55
R	0	3	419	2	37		R	0.00	0.65	90.89	0.43	8.03
U	0	2	2	30	2		U	0.00	5.56	5.56	83.33	5.56
N	4	251	417	79	5222		N	0.07	4.20	6.98	1.32	87.43

**TAN** Acierto

	D	P	R	U	N		D	P	R	U	N	
D	3745	3	0	0	0	34	D	99.02	0.08	0.00	0.00	0.90
P	1	723	0	0	0	8	P	0.14	98.77	0.00	0.00	1.09
R	0	1	434	0	0	26	R	0.00	0.22	94.14	0.00	5.64
U	0	0	3	32	1		U	0.00	0.00	8.33	88.89	2.78
N	12	21	24	3	5913		N	0.20	0.35	0.40	0.05	99.00

**RBFNet** Acierto

	D	P	R	U	N		D	P	R	U	N	
D	3542	24	12	0	0	204	D	93.65	0.63	0.32	0.00	5.39
P	3	710	2	0	0	17	P	0.41	96.99	0.27	0.00	2.32
R	0	2	373	1	0	85	R	0.00	0.43	80.91	0.22	18.44
U	0	5	1	21	0	9	U	0.00	13.89	2.78	58.33	25.00
N	17	101	93	12	5750		N	0.28	1.69	1.56	0.20	96.27

**KNN-1** Acierto

	D	P	R	U	N		D	P	R	U	N	
D	3780	0	0	0	0	2	D	99.95	0.00	0.00	0.00	0.05
P	0	732	0	0	0	0	P	0.00	100.00	0.00	0.00	0.00
R	0	0	448	0	0	13	R	0.00	0.00	97.18	0.00	2.82
U	0	0	0	36	0		U	0.00	0.00	0.00	100.00	0.00
N	0	0	0	0	0	5973	N	0.00	0.00	0.00	0.00	100.00

**KNN-50** Acierto

	D	P	R	U	N		D	P	R	U	N	
D	3766	8	0	0	0	8	D	99.58	0.21	0.00	0.00	0.21
P	55	638	4	1	0	34	P	7.51	87.16	0.55	0.14	4.64

R	0	1	391	0	69	R	0.00	0.22	84.82	0.00	14.97
U	0	6	2	14	14	U	0.00	16.67	5.56	38.89	38.89
N	39	75	23	4	5832	N	0.65	1.26	0.39	0.07	97.64

## FrecCFS

Clonal						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3091	0	0	0	691	D	81.73	0.00	0.00	0.00	18.27
P		281	0	0	0	451	P	38.39	0.00	0.00	0.00	61.61
R		2	0	0	0	459	R	0.43	0.00	0.00	0.00	99.57
U		6	0	0	0	30	U	16.67	0.00	0.00	0.00	83.33
N		45	0	0	0	5928	N	0.75	0.00	0.00	0.00	99.25

Genetico						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3146	0	0	0	636	D	83.18	0.00	0.00	0.00	16.82
P		400	0	0	0	332	P	54.64	0.00	0.00	0.00	45.36
R		19	0	74	0	368	R	4.12	0.00	16.05	0.00	79.83
U		6	0	0	0	30	U	16.67	0.00	0.00	0.00	83.33
N		237	0	5	0	5731	N	3.97	0.00	0.08	0.00	95.95

Furia						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3778	2	0	0	2	D	99.89	0.05	0.00	0.00	0.05
P		24	681	0	0	27	P	3.28	93.03	0.00	0.00	3.69
R		0	0	311	0	150	R	0.00	0.00	67.46	0.00	32.54
U		0	6	0	19	11	U	0.00	16.67	0.00	52.78	30.56
N		16	25	7	2	5923	N	0.27	0.42	0.12	0.03	99.16

Part						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3782	2	0	0	7	D	99.76	0.05	0.00	0.00	0.18
P		732	714	0	0	14	P	50.14	48.90	0.00	0.00	0.96
R		461	0	404	1	55	R	50.05	0.00	43.87	0.11	5.97
U		36	5	0	25	6	U	50.00	6.94	0.00	34.72	8.33
N		5638	17	22	4	5913	N	48.63	0.15	0.19	0.03	51.00

Ripper						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3773	8	0	0	1	D	99.76	0.21	0.00	0.00	0.03
P		2	713	0	0	17	P	0.27	97.40	0.00	0.00	2.32
R		0	0	347	0	114	R	0.00	0.00	75.27	0.00	24.73
U		0	1	0	29	6	U	0.00	2.78	0.00	80.56	16.67
N		17	54	51	3	5848	N	0.28	0.90	0.85	0.05	97.91

RNDF						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3780	1	0	0	1	D	99.95	0.03	0.00	0.00	0.03
P		1	724	0	0	7	P	0.14	98.91	0.00	0.00	0.96
R		0	0	427	0	34	R	0.00	0.00	92.62	0.00	7.38
U		0	0	0	34	2	U	0.00	0.00	0.00	94.44	5.56
N		15	3	25	2	5928	N	0.25	0.05	0.42	0.03	99.25

C4.5						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3777	4	0	0	1	D	99.87	0.11	0.00	0.00	0.03
P		4	716	0	0	12	P	0.55	97.81	0.00	0.00	1.64
R		1	0	400	0	60	R	0.22	0.00	86.77	0.00	13.02
U		0	6	2	20	8	U	0.00	16.67	5.56	55.56	22.22
N		17	47	9	2	5898	N	0.28	0.79	0.15	0.03	98.74

NBTree						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3770	6	0	0	6	D	99.68	0.16	0.00	0.00	0.16
P		7	710	0	0	15	P	0.96	96.99	0.00	0.00	2.05
R		1	0	407	0	53	R	0.22	0.00	88.29	0.00	11.50
U		0	6	0	5	25	U	0.00	16.67	0.00	13.89	69.44
N		22	27	33	1	5890	N	0.37	0.45	0.55	0.02	98.61

CART						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3778	1	0	0	3	D	99.89	0.03	0.00	0.00	0.08
P		3	711	0	0	18	P	0.41	97.13	0.00	0.00	2.46
R		0	1	413	1	46	R	0.00	0.22	89.59	0.22	9.98
U		0	0	0	29	7	U	0.00	0.00	0.00	80.56	19.44
N		17	10	24	4	5918	N	0.28	0.17	0.40	0.07	99.08

SMOPoly						Acierto					
	D	P	R	U	N		D	P	R	U	N

D	3772	1	0	0	9	D	99.74	0.03	0.00	0.00	0.24
P	2	709	0	0	21	P	0.27	96.86	0.00	0.00	2.87
R	0	3	405	0	53	R	0.00	0.65	87.85	0.00	11.50
U	0	0	0	30	6	U	0.00	0.00	0.00	83.33	16.67
N	22	32	27	3	5889	N	0.37	0.54	0.45	0.05	98.59

**SMORBF** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3729	2	0	0	51	D	98.60	0.05	0.00	0.00	1.35
P	29	683	0	0	20	P	3.96	93.31	0.00	0.00	2.73
R	0	2	305	0	154	R	0.00	0.43	66.16	0.00	33.41
U	1	21	0	0	14	U	2.78	58.33	0.00	0.00	38.89
N	40	56	5	0	5872	N	0.67	0.94	0.08	0.00	98.31

**CSV-RBF** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3778	1	0	0	3	D	99.89	0.03	0.00	0.00	0.08
P	1	722	0	0	9	P	0.14	98.63	0.00	0.00	1.23
R	0	1	414	0	46	R	0.00	0.22	89.80	0.00	9.98
U	0	0	1	29	6	U	0.00	0.00	2.78	80.56	16.67
N	19	21	23	2	5908	N	0.32	0.35	0.39	0.03	98.91

**CSV-SIGM** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	0	0	3	0	3779	D	0.00	0.00	0.08	0.00	99.92
P	0	0	0	0	732	P	0.00	0.00	0.00	0.00	100.00
R	0	0	0	0	461	R	0.00	0.00	0.00	0.00	100.00
U	0	0	0	0	36	U	0.00	0.00	0.00	0.00	100.00
N	0	0	0	0	5973	N	0.00	0.00	0.00	0.00	100.00

**PML** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3590	2	0	0	190	D	94.92	0.05	0.00	0.00	5.02
P	3	711	0	0	18	P	0.41	97.13	0.00	0.00	2.46
R	1	1	0	0	459	R	0.22	0.22	0.00	0.00	99.57
U	0	6	0	0	30	U	0.00	16.67	0.00	0.00	83.33
N	13	47	0	0	5913	N	0.22	0.79	0.00	0.00	99.00

**NaiveBayes** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3325	87	43	0	327	D	87.92	2.30	1.14	0.00	8.65
P	34	684	1	0	13	P	4.64	93.44	0.14	0.00	1.78
R	0	4	418	1	38	R	0.00	0.87	90.67	0.22	8.24
U	0	21	0	5	10	U	0.00	58.33	0.00	13.89	27.78
N	39	122	211	39	5562	N	0.65	2.04	3.53	0.65	93.12

**TAN** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3637	84	0	0	61	D	96.17	2.22	0.00	0.00	1.61
P	19	694	0	0	19	P	2.60	94.81	0.00	0.00	2.60
R	0	0	410	0	51	R	0.00	0.00	88.94	0.00	11.06
U	0	0	1	28	7	U	0.00	0.00	2.78	77.78	19.44
N	6	28	27	4	5908	N	0.10	0.47	0.45	0.07	98.91

**RBFNet** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3715	13	0	0	54	D	98.23	0.34	0.00	0.00	1.43
P	19	673	0	0	40	P	2.60	91.94	0.00	0.00	5.46
R	1	0	380	0	80	R	0.22	0.00	82.43	0.00	17.35
U	0	6	0	22	8	U	0.00	16.67	0.00	61.11	22.22
N	42	81	62	6	5782	N	0.70	1.36	1.04	0.10	96.80

**KNN-1** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3780	1	0	0	1	D	99.95	0.03	0.00	0.00	0.03
P	1	721	0	0	10	P	0.14	98.50	0.00	0.00	1.37
R	0	0	425	0	36	R	0.00	0.00	92.19	0.00	7.81
U	0	0	0	34	2	U	0.00	0.00	0.00	94.44	5.56
N	15	2	23	2	5931	N	0.25	0.03	0.39	0.03	99.30

**KNN-50** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3780	1	0	0	1	D	99.95	0.03	0.00	0.00	0.03
P	54	620	14	0	44	P	7.38	84.70	1.91	0.00	6.01
R	2	2	373	1	83	R	0.43	0.43	80.91	0.22	18.00

U	0	5	1	1	29
N	70	79	43	0	5781

U	0.00	13.89	2.78	2.78	80.56
N	1.17	1.32	0.72	0.00	96.79



## FrecCNS

Clonal							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		2924	0	0	0	858	D	77.31	0.00	0.00	0.00	22.69
P		142	0	0	0	590	P	19.40	0.00	0.00	0.00	80.60
R		1	0	0	0	460	R	0.22	0.00	0.00	0.00	99.78
U		0	0	0	0	36	U	0.00	0.00	0.00	0.00	100.00
N		408	0	0	0	5565	N	6.83	0.00	0.00	0.00	93.17

Genetico							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		3172	60	45	14	491	D	83.87	1.59	1.19	0.37	12.98
P		247	90	11	3	381	P	33.74	12.30	1.50	0.41	52.05
R		1	16	36	1	407	R	0.22	3.47	7.81	0.22	88.29
U		14	1	0	1	20	U	38.89	2.78	0.00	2.78	55.56
N		41	60	19	13	5840	N	0.69	1.00	0.32	0.22	97.77

Furia							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		3776	0	0	0	6	D	99.84	0.00	0.00	0.00	0.16
P		0	727	0	0	5	P	0.00	99.32	0.00	0.00	0.68
R		1	1	341	0	118	R	0.22	0.22	73.97	0.00	25.60
U		0	2	1	21	12	U	0.00	5.56	2.78	58.33	33.33
N		14	16	2	0	5941	N	0.23	0.27	0.03	0.00	99.46

Part							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		3755	9	0	0	18	D	99.29	0.24	0.00	0.00	0.48
P		8	710	0	0	14	P	1.09	96.99	0.00	0.00	1.91
R		0	0	419	0	42	R	0.00	0.00	90.89	0.00	9.11
U		0	0	0	27	9	U	0.00	0.00	0.00	75.00	25.00
N		13	10	14	1	5935	N	0.22	0.17	0.23	0.02	99.36

Ripper							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		3775	3	0	0	4	D	99.81	0.08	0.00	0.00	0.11
P		0	718	1	0	13	P	0.00	98.09	0.14	0.00	1.78
R		0	0	404	0	57	R	0.00	0.00	87.64	0.00	12.36
U		0	0	1	27	8	U	0.00	0.00	2.78	75.00	22.22
N		13	20	36	3	5901	N	0.22	0.33	0.60	0.05	98.79

RNDF							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		3781	0	0	0	1	D	99.97	0.00	0.00	0.00	0.03
P		0	732	0	0	0	P	0.00	100.00	0.00	0.00	0.00
R		0	0	452	0	9	R	0.00	0.00	98.05	0.00	1.95
U		0	0	0	36	0	U	0.00	0.00	0.00	100.00	0.00
N		2	0	3	0	5968	N	0.03	0.00	0.05	0.00	99.92

C4.5							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		3762	3	0	0	17	D	99.47	0.08	0.00	0.00	0.45
P		3	717	2	0	9	P	0.41	98.08	0.27	0.00	1.23
R		0	0	412	0	49	R	0.00	0.00	89.37	0.00	10.63
U		0	5	0	27	4	U	0.00	13.89	0.00	75.00	11.11
N		14	20	10	3	5926	N	0.23	0.33	0.17	0.05	99.21

NBTree							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		3777	3	0	0	2	D	99.87	0.08	0.00	0.00	0.05
P		3	725	0	0	4	P	0.41	99.04	0.00	0.00	0.55
R		0	1	415	0	45	R	0.00	0.22	90.02	0.00	9.76
U		1	4	0	22	9	U	2.78	11.11	0.00	61.11	25.00
N		16	18	5	3	5931	N	0.27	0.30	0.08	0.05	99.30

CART							Acierto					
	D	P	R	U	N		D	P	R	U	N	
D		3774	0	0	0	8	D	99.79	0.00	0.00	0.00	0.21
P		1	721	0	0	10	P	0.14	98.50	0.00	0.00	1.37
R		0	1	426	0	34	R	0.00	0.22	92.41	0.00	7.38
U		0	1	0	30	5	U	0.00	2.78	0.00	83.33	13.89
N		8	5	3	1	5956	N	0.13	0.08	0.05	0.02	99.72

SMOPoly							Acierto					
	D	P	R	U	N		D	P	R	U	N	

D	3780	1	0	0	1	D	99.95	0.03	0.00	0.00	0.03
P	0	727	0	0	5	P	0.00	99.32	0.00	0.00	0.68
R	0	0	420	0	41	R	0.00	0.00	91.11	0.00	8.89
U	0	0	0	34	2	U	0.00	0.00	0.00	94.44	5.56
N	9	15	3	0	5946	N	0.15	0.25	0.05	0.00	99.55

**SMORBF** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3758	1	0	0	23	D	99.37	0.03	0.00	0.00	0.61
P	4	711	2	0	15	P	0.55	97.13	0.27	0.00	2.05
R	0	2	380	0	79	R	0.00	0.43	82.43	0.00	17.14
U	0	4	1	0	31	U	0.00	11.11	2.78	0.00	86.11
N	24	55	5	0	5889	N	0.40	0.92	0.08	0.00	98.59

**CSV-RBF** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3779	0	0	0	3	D	99.92	0.00	0.00	0.00	0.08
P	0	731	0	0	1	P	0.00	99.86	0.00	0.00	0.14
R	0	0	440	0	21	R	0.00	0.00	95.44	0.00	4.56
U	0	0	0	33	3	U	0.00	0.00	0.00	91.67	8.33
N	3	2	6	0	5962	N	0.05	0.03	0.10	0.00	99.82

**CSV-SIGM** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	14	0	0	0	3768	D	0.37	0.00	0.00	0.00	99.63
P	0	0	0	0	732	P	0.00	0.00	0.00	0.00	100.00
R	0	0	0	0	461	R	0.00	0.00	0.00	0.00	100.00
U	0	0	0	0	36	U	0.00	0.00	0.00	0.00	100.00
N	1	0	0	0	5972	N	0.02	0.00	0.00	0.00	99.98

**PML** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	0	23	0	0	14	D	0.00	62.16	0.00	0.00	37.84
P	2	717	0	2	11	P	0.27	97.95	0.00	0.27	1.50
R	348	7	0	348	106	R	43.02	0.87	0.00	43.02	13.10
U	14	12	0	14	10	U	28.00	24.00	0.00	28.00	20.00
N	3	91	0	3	5865	N	0.05	1.53	0.00	0.05	98.37

**NaiveBayes** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3243	83	12	84	360	D	85.75	2.19	0.32	2.22	9.52
P	1	707	9	13	2	P	0.14	96.58	1.23	1.78	0.27
R	0	2	423	4	32	R	0.00	0.43	91.76	0.87	6.94
U	0	1	1	32	2	U	0.00	2.78	2.78	88.89	5.56
N	11	146	364	318	5134	N	0.18	2.44	6.09	5.32	85.95

**TAN** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3750	2	0	0	30	D	99.15	0.05	0.00	0.00	0.79
P	1	719	0	0	12	P	0.14	98.22	0.00	0.00	1.64
R	0	2	434	0	25	R	0.00	0.43	94.14	0.00	5.42
U	0	3	3	25	5	U	0.00	8.33	8.33	69.44	13.89
N	13	48	29	1	5882	N	0.22	0.80	0.49	0.02	98.48

**RBFNet** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3623	29	1	2	127	D	95.80	0.77	0.03	0.05	3.36
P	3	713	1	0	15	P	0.41	97.40	0.14	0.00	2.05
R	0	1	413	0	47	R	0.00	0.22	89.59	0.00	10.20
U	2	3	1	24	6	U	5.56	8.33	2.78	66.67	16.67
N	37	86	140	9	5701	N	0.62	1.44	2.34	0.15	95.45

**KNN-1** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3780	0	0	0	2	D	99.95	0.00	0.00	0.00	0.05
P	0	732	0	0	0	P	0.00	100.00	0.00	0.00	0.00
R	0	0	449	0	12	R	0.00	0.00	97.40	0.00	2.60
U	0	0	0	36	0	U	0.00	0.00	0.00	100.00	0.00
N	0	0	0	0	5973	N	0.00	0.00	0.00	0.00	100.00

**KNN-50** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3742	5	0	0	35	D	98.94	0.13	0.00	0.00	0.93
P	25	684	3	0	20	P	3.42	93.44	0.41	0.00	2.73
R	0	1	402	0	58	R	0.00	0.22	87.20	0.00	12.58

U	0	4	2	15	15	U	0.00	11.11	5.56	41.67	41.67
N	32	53	30	0	5858	N	0.54	0.89	0.50	0.00	98.07

## FrecC4.5

Clonalg						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3035	0	11	0	736	D	80.25	0.00	0.29	0.00	19.46
P		265	41	29	0	397	P	36.20	5.60	3.96	0.00	54.23
R		4	0	37	0	420	R	0.87	0.00	8.03	0.00	91.11
U		6	0	5	0	25	U	16.67	0.00	13.89	0.00	69.44
N		25	1	24	0	5923	N	0.42	0.02	0.40	0.00	99.16

Genetico						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3114	21	0	4	643	D	82.34	0.56	0.00	0.11	17.00
P		87	367	0	22	256	P	11.89	50.14	0.00	3.01	34.97
R		1	3	74	0	383	R	0.22	0.65	16.05	0.00	83.08
U		1	13	0	9	13	U	2.78	36.11	0.00	25.00	36.11
N		57	268	7	5	5636	N	0.95	4.49	0.12	0.08	94.36

Furia						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3779	0	0	0	3	D	99.92	0.00	0.00	0.00	0.08
P		11	705	0	0	16	P	1.50	96.31	0.00	0.00	2.19
R		0	0	360	0	101	R	0.00	0.00	78.09	0.00	21.91
U		0	1	2	28	5	U	0.00	2.78	5.56	77.78	13.89
N		37	36	10	1	5889	N	0.62	0.60	0.17	0.02	98.59

Part						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3768	2	0	0	12	D	99.63	0.05	0.00	0.00	0.32
P		8	692	1	0	31	P	1.09	94.54	0.14	0.00	4.23
R		1	0	402	0	58	R	0.22	0.00	87.20	0.00	12.58
U		0	0	2	26	8	U	0.00	0.00	5.56	72.22	22.22
N		15	19	25	1	5913	N	0.25	0.32	0.42	0.02	99.00

Ripper						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3775	6	0	0	1	D	99.81	0.16	0.00	0.00	0.03
P		2	710	0	1	19	P	0.27	96.99	0.00	0.14	2.60
R		0	0	420	2	39	R	0.00	0.00	91.11	0.43	8.46
U		0	0	0	31	5	U	0.00	0.00	0.00	86.11	13.89
N		15	38	45	1	5874	N	0.25	0.64	0.75	0.02	98.34

RNDF						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3780	0	0	0	2	D	99.95	0.00	0.00	0.00	0.05
P		0	732	0	0	0	P	0.00	100.00	0.00	0.00	0.00
R		0	0	451	0	10	R	0.00	0.00	97.83	0.00	2.17
U		0	0	0	36	0	U	0.00	0.00	0.00	100.00	0.00
N		2	1	8	0	5962	N	0.03	0.02	0.13	0.00	99.82

C4.5						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3775	4	0	0	3	D	99.81	0.11	0.00	0.00	0.08
P		5	690	1	0	36	P	0.68	94.26	0.14	0.00	4.92
R		1	0	407	1	52	R	0.22	0.00	88.29	0.22	11.28
U		0	6	2	24	4	U	0.00	16.67	5.56	66.67	11.11
N		27	21	24	6	5895	N	0.45	0.35	0.40	0.10	98.69

NBTREE						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3772	0	1	0	9	D	99.74	0.00	0.03	0.00	0.24
P		29	686	0	0	17	P	3.96	93.72	0.00	0.00	2.32
R		0	1	416	1	43	R	0.00	0.22	90.24	0.22	9.33
U		0	0	2	32	2	U	0.00	0.00	5.56	88.89	5.56
N		25	16	26	3	5903	N	0.42	0.27	0.44	0.05	98.83

CART						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3772	1	0	0	9	D	99.74	0.03	0.00	0.00	0.24
P		3	716	0	0	13	P	0.41	97.81	0.00	0.00	1.78
R		1	1	431	1	27	R	0.22	0.22	93.49	0.22	5.86
U		0	0	0	33	3	U	0.00	0.00	0.00	91.67	8.33
N		7	9	17	2	5938	N	0.12	0.15	0.28	0.03	99.41

SMOPoly						Acierto					
	D	P	R	U	N		D	P	R	U	N

D	3776	0	0	0	6	D	99.84	0.00	0.00	0.00	0.16
P	0	721	0	0	11	P	0.00	98.50	0.00	0.00	1.50
R	0	0	429	0	32	R	0.00	0.00	93.06	0.00	6.94
U	0	0	0	35	1	U	0.00	0.00	0.00	97.22	2.78
N	8	18	24	0	5923	N	0.13	0.30	0.40	0.00	99.16

**SMORBF** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3774	1	0	0	7	D	99.79	0.03	0.00	0.00	0.19
P	1	690	1	0	40	P	0.14	94.26	0.14	0.00	5.46
R	0	2	350	0	109	R	0.00	0.43	75.92	0.00	23.64
U	0	7	1	10	18	U	0.00	19.44	2.78	27.78	50.00
N	30	43	9	0	5891	N	0.50	0.72	0.15	0.00	98.63

**CSV-RBF** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3771	7	0	0	3782	D	49.88	0.09	0.00	0.00	50.03
P	1	728	0	0	732	P	0.07	49.83	0.00	0.00	50.10
R	0	0	435	0	461	R	0.00	0.00	48.55	0.00	51.45
U	0	0	1	29	36	U	0.00	0.00	1.52	43.94	54.55
N	4	4	17	0	5973	N	0.07	0.07	0.28	0.00	99.58

**CSV-SIGM** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	0	0	0	0	3782	D	0.00	0.00	0.00	0.00	100.00
P	0	0	0	0	732	P	0.00	0.00	0.00	0.00	100.00
R	0	0	0	0	461	R	0.00	0.00	0.00	0.00	100.00
U	0	0	0	0	36	U	0.00	0.00	0.00	0.00	100.00
N	0	0	0	0	5973	N	0.00	0.00	0.00	0.00	100.00

**PML** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3772	3	1	0	6	D	99.74	0.08	0.03	0.00	0.16
P	4	712	2	0	14	P	0.55	97.27	0.27	0.00	1.91
R	0	8	383	0	70	R	0.00	1.74	83.08	0.00	15.18
U	0	19	1	0	16	U	0.00	52.78	2.78	0.00	44.44
N	20	105	22	0	5826	N	0.33	1.76	0.37	0.00	97.54

**Naive Bayes** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3114	202	61	20	385	D	82.34	5.34	1.61	0.53	10.18
P	5	700	13	9	5	P	0.68	95.63	1.78	1.23	0.68
R	0	5	414	2	40	R	0.00	1.08	89.80	0.43	8.68
U	0	3	3	28	2	U	0.00	8.33	8.33	77.78	5.56
N	27	289	287	75	5295	N	0.45	4.84	4.80	1.26	88.65

**TAN** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3748	2	2	0	30	D	99.10	0.05	0.05	0.00	0.79
P	3	714	1	0	14	P	0.41	97.54	0.14	0.00	1.91
R	0	0	415	1	45	R	0.00	0.00	90.02	0.22	9.76
U	0	0	2	30	4	U	0.00	0.00	5.56	83.33	11.11
N	23	29	51	1	5869	N	0.39	0.49	0.85	0.02	98.26

**RBFNet** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3711	22	0	0	49	D	98.12	0.58	0.00	0.00	1.30
P	1	703	1	2	25	P	0.14	96.04	0.14	0.27	3.42
R	0	2	387	1	71	R	0.00	0.43	83.95	0.22	15.40
U	0	4	4	25	3	U	0.00	11.11	11.11	69.44	8.33
N	28	115	97	6	5727	N	0.47	1.93	1.62	0.10	95.88

**KNN-1** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3779	0	0	0	3	D	99.92	0.00	0.00	0.00	0.08
P	0	731	0	0	1	P	0.00	99.86	0.00	0.00	0.14
R	0	0	437	0	13	R	0.00	0.00	97.11	0.00	2.89
U	0	0	0	36	0	U	0.00	0.00	0.00	100.00	0.00
N	0	0	2	0	5969	N	0.00	0.00	0.03	0.00	99.97

**KNN-50** **Acierto**

	D	P	R	U	N		D	P	R	U	N
D	3748	13	10	0	11	D	99.10	0.34	0.26	0.00	0.29
P	29	656	6	2	39	P	3.96	89.62	0.82	0.27	5.33
R	0	1	374	0	86	R	0.00	0.22	81.13	0.00	18.66

U	8	4	2	7	15	U	22.22	11.11	5.56	19.44	41.67
N	44	70	41	3	5815	N	0.74	1.17	0.69	0.05	97.35

## FrecNB

Clonalg						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3022	0	7	0	753	D	79.90	0.00	0.19	0.00	19.91
P		167	34	105	0	426	P	22.81	4.64	14.34	0.00	58.20
R		2	0	20	0	439	R	0.43	0.00	4.34	0.00	95.23
U		7	0	4	0	25	U	19.44	0.00	11.11	0.00	69.44
N		10	0	61	0	5902	N	0.17	0.00	1.02	0.00	98.81

Genetico						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3222	0	0	0	560	D	85.19	0.00	0.00	0.00	14.81
P		441	62	0	0	229	P	60.25	8.47	0.00	0.00	31.28
R		18	6	114	0	323	R	3.90	1.30	24.73	0.00	70.07
U		21	0	0	3	12	U	58.33	0.00	0.00	8.33	33.33
N		335	109	49	0	5480	N	5.61	1.82	0.82	0.00	91.75

Furia						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3593	18	0	0	171	D	95.00	0.48	0.00	0.00	4.52
P		88	474	4	0	166	P	12.02	64.75	0.55	0.00	22.68
R		1	0	302	1	157	R	0.22	0.00	65.51	0.22	34.06
U		1	5	1	20	9	U	2.78	13.89	2.78	55.56	25.00
N		65	51	39	2	5816	N	1.09	0.85	0.65	0.03	97.37

Part						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3616	2	0	0	164	D	95.61	0.05	0.00	0.00	4.34
P		16	653	6	0	57	P	2.19	89.21	0.82	0.00	7.79
R		1	4	299	0	157	R	0.22	0.87	64.86	0.00	34.06
U		2	5	2	18	9	U	5.56	13.89	5.56	50.00	25.00
N		51	45	45	2	5830	N	0.85	0.75	0.75	0.03	97.61

Ripper						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3572	1	0	0	209	D	94.45	0.03	0.00	0.00	5.53
P		14	489	3	0	226	P	1.91	66.80	0.41	0.00	30.87
R		0	0	289	2	170	R	0.00	0.00	62.69	0.43	36.88
U		1	5	1	23	6	U	2.78	13.89	2.78	63.89	16.67
N		45	39	41	4	5844	N	0.75	0.65	0.69	0.07	97.84

RNDF						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3608	5	0	0	169	D	95.40	0.13	0.00	0.00	4.47
P		8	674	3	0	47	P	1.09	92.08	0.41	0.00	6.42
R		0	3	341	0	117	R	0.00	0.65	73.97	0.00	25.38
U		1	5	2	27	1	U	2.78	13.89	5.56	75.00	2.78
N		36	29	57	0	5851	N	0.60	0.49	0.95	0.00	97.96

C4.5						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3612	5	0	0	165	D	95.51	0.13	0.00	0.00	4.36
P		15	657	3	0	57	P	2.05	89.75	0.41	0.00	7.79
R		1	2	331	2	125	R	0.22	0.43	71.80	0.43	27.11
U		2	6	2	25	1	U	5.56	16.67	5.56	69.44	2.78
N		48	42	54	4	5825	N	0.80	0.70	0.90	0.07	97.52

NBTREE						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3605	11	0	0	166	D	95.32	0.29	0.00	0.00	4.39
P		42	631	3	0	56	P	5.74	86.20	0.41	0.00	7.65
R		1	2	331	2	125	R	0.22	0.43	71.80	0.43	27.11
U		1	6	2	24	3	U	2.78	16.67	5.56	66.67	8.33
N		49	53	53	4	5814	N	0.82	0.89	0.89	0.07	97.34

CART						Acierto						
	D	P	R	U	N		D	P	R	U	N	
D		3614	3	0	0	165	D	95.56	0.08	0.00	0.00	4.36
P		21	644	3	0	64	P	2.87	87.98	0.41	0.00	8.74
R		1	3	314	1	142	R	0.22	0.65	68.11	0.22	30.80
U		2	5	2	19	8	U	5.56	13.89	5.56	52.78	22.22
N		51	41	47	0	5834	N	0.85	0.69	0.79	0.00	97.67

SMOPoly						Acierto					
	D	P	R	U	N		D	P	R	U	N

D	3610	4	0	0	168	D	95.45	0.11	0.00	0.00	4.44
P	18	646	3	0	65	P	2.46	88.25	0.41	0.00	8.88
R	0	2	337	1	121	R	0.00	0.43	73.10	0.22	26.25
U	1	5	2	20	8	U	2.78	13.89	5.56	55.56	22.22
N	49	57	74	1	5792	N	0.82	0.95	1.24	0.02	96.97

SMORBF						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	3521	2	0	0	259	D	93.10	0.05	0.00	0.00	6.85
P	21	578	1	0	132	P	2.87	78.96	0.14	0.00	18.03
R	4	2	259	0	196	R	0.87	0.43	56.18	0.00	42.52
U	0	21	0	0	15	U	0.00	58.33	0.00	0.00	41.67
N	38	66	58	0	5811	N	0.64	1.10	0.97	0.00	97.29

CSV-RBF						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	3596	8	0	0	178	D	95.08	0.21	0.00	0.00	4.71
P	11	654	2	0	65	P	1.50	89.34	0.27	0.00	8.88
R	0	4	313	0	144	R	0.00	0.87	67.90	0.00	31.24
U	2	6	2	15	11	U	5.56	16.67	5.56	41.67	30.56
N	53	31	49	0	5840	N	0.89	0.52	0.82	0.00	97.77

CSV-SIGM						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	0	0	0	0	3782	D	0.00	0.00	0.00	0.00	100.00
P	0	0	0	0	732	P	0.00	0.00	0.00	0.00	100.00
R	0	1	0	0	460	R	0.00	0.22	0.00	0.00	99.78
U	0	0	0	0	36	U	0.00	0.00	0.00	0.00	100.00
N	2	1	0	0	5970	N	0.03	0.02	0.00	0.00	99.95

PML						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	3611	3	0	0	168	D	95.48	0.08	0.00	0.00	4.44
P	13	646	12	0	61	P	1.78	88.25	1.64	0.00	8.33
R	1	3	250	0	207	R	0.22	0.65	54.23	0.00	44.90
U	2	21	0	0	13	U	5.56	58.33	0.00	0.00	36.11
N	43	43	24	0	5863	N	0.72	0.72	0.40	0.00	98.16

NaiveBayes						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	3294	8	0	0	480	D	87.10	0.21	0.00	0.00	12.69
P	71	623	2	2	34	P	9.70	85.11	0.27	0.27	4.64
R	0	17	265	1	178	R	0.00	3.69	57.48	0.22	38.61
U	0	16	0	9	11	U	0.00	44.44	0.00	25.00	30.56
N	14	184	66	2	5707	N	0.23	3.08	1.10	0.03	95.55

TAN						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	3560	15	0	0	207	D	94.13	0.40	0.00	0.00	5.47
P	22	616	3	7	84	P	3.01	84.15	0.41	0.96	11.48
R	0	2	257	1	201	R	0.00	0.43	55.75	0.22	43.60
U	1	5	0	20	10	U	2.78	13.89	0.00	55.56	27.78
N	37	35	47	5	5849	N	0.62	0.59	0.79	0.08	97.92

RBFNet						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	3413	52	2	0	315	D	90.24	1.37	0.05	0.00	8.33
P	43	614	9	0	66	P	5.87	83.88	1.23	0.00	9.02
R	0	3	335	1	122	R	0.00	0.65	72.67	0.22	26.46
U	0	6	2	19	9	U	0.00	16.67	5.56	52.78	25.00
N	16	66	103	1	5787	N	0.27	1.10	1.72	0.02	96.89

KNN-1						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	3610	2	0	0	170	D	95.45	0.05	0.00	0.00	4.49
P	10	671	2	0	49	P	1.37	91.67	0.27	0.00	6.69
R	0	4	335	0	122	R	0.00	0.87	72.67	0.00	26.46
U	1	5	2	26	2	U	2.78	13.89	5.56	72.22	5.56
N	34	26	48	0	5865	N	0.57	0.44	0.80	0.00	98.19

KNN-50						Acierto					
	D	P	R	U	N		D	P	R	U	N
D	3604	2	0	0	176	D	95.29	0.05	0.00	0.00	4.65
P	115	424	3	0	190	P	15.71	57.92	0.41	0.00	25.96
R	3	0	246	0	212	R	0.65	0.00	53.36	0.00	45.99



U	2	16	0	2	16	U	5.56	44.44	0.00	5.56	44.44
N	73	38	57	2	5803	N	1.22	0.64	0.95	0.03	97.15

Matrices de Confusión para el Tercer  
Estudio a nivel de 20 Categorías de  
Ataques









































