Departamento de Arquitectura y Tecnología de Computadores

E.T.S. Ingeniería Informática y de Telecomunicación

Doctorado en Tecnologías de la Información y la Comunicación

Universidad de Granada

TESIS DOCTORAL

"Mapas auto-organizativos probabilísticos y análisis en componentes de conexiones para la detección de anomalías en redes de computadores"

Autor
Eduardo Miguel De la Hoz Correa

Directores
Dr. Andrés Ortiz García
Dr. Julio Ortega Lopera

Editor: Universidad de Granada. Tesis Doctorales
Autor: Eduardo Miguel de la Hoz Correa
ISBN: 978-84-9125-981-7
URI: http://hdl.handle.net/10481/44083

El doctorando Eduardo Miguel De la Hoz Correa y los directores de la tesis Dr. Andrés Ortiz García y Dr. Julio Ortega Lopera garantizamos, al firmar esta tesis doctoral, que el trabajo ha sido realizado por el doctorando bajo la dirección de los directores de la tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo, se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

Granada, abril 19 de 2016

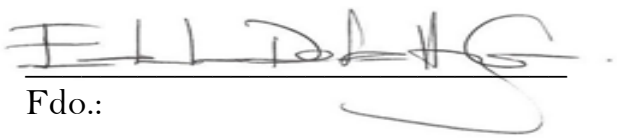
Dr. Andrés Ortiz García
Director de la Tesis

Fdo.:

Julio Ortega Lopera
Dr. Director de la Tesis

Fdo.:

Eduardo Miguel De la Hoz Correa
Doctorando



Fdo.:

Agradecimientos

A Dios,
Señor y dador de vida por dejarme iniciar y terminar todos mis proyectos, gracias Padre celestial

A mis padres,
Por su apoyo incondicional y ayuda en todos los aspectos de mi vida, por guiarme a ser el hombre que hoy he llegado a ser

A mi esposa Dayana,
Por ser mi compañera, comprenderme y soportar a mi lado las pruebas de la vida junto a mí.

A mi pequeño Eduardo Jesús,
Para que día a día comprenda que con amor, perseverancia y ganas todo es posible en esta vida

A mis directores de tesis, Julio y Andrés,
Profesionales que no solo se encargaron en direccionarme académicamente esta Tesis, sino que se han convertido en mis mayores referentes a seguir como personas, como apoyos incansables en la búsqueda del conocimiento, por quienes sin duda alguna este proyecto ha llegado a feliz término, muchas gracias.

A la Universidad de la Costa – CUC,
Por su apoyo económicamente en todo el proceso de este estudio y su confianza depositada en esta labor.

A la Universidad de Granada – UGR,
Y a todos los docentes del master en Ingeniería de Computadores y Redes y el doctorado en Tecnologías de la Información y Comunicación – TIC, por darme las bases del conocimiento y el afán por la búsqueda del mismo.

Al Ministerio de Economía y Competitividad de España y de los Fondos Feder, que han financiado los proyecto TIN2012-32039, TIN2015-67020-P, y PSI2015-65848-R, en el cual se enmarca este trabajo

A todos muchísimas gracias.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1. Presentación y objetivos de la Tesis

El crecimiento de Internet y, en consecuencia, el número de ordenadores interconectados, ha expuesto a cantidades significativas de información a los intrusos y atacantes. Los cortafuegos también llamados *Firewalls* tienen como objetivo detectar violaciones de acuerdo con un conjunto de reglas predefinidas y generalmente bloquean el tráfico entrante potencialmente peligroso. Sin embargo, con la evolución de las técnicas de ataque, es más difícil distinguir las anomalías del tráfico normal.

Se han propuesto diferentes enfoques de detección, incluyendo el uso de técnicas de aprendizaje automático basado en modelos neuronales, tales como *Self-Organizing Maps* (*SOM*). En la presente tesis, presentamos un enfoque de clasificación donde se hibridan técnicas estadísticas y *SOM* para la detección de anomalías de red las cuales han dado resultados prometedores, y cuya aplicación se ha testeado con resultados muy satisfactorios. Se utilizó *Principal Component Analysis* (*PCA*) y *Fisher Discriminant Ratio* (*FDR*) para la selección de características y eliminación de ruido, mientras que los *Probabilistic Self-Organizing Maps* (*PSOM*) tienen como objetivo modelar el espacio de características y permitir distinguir entre las conexiones normales y anómalas. Las capacidades de detección del sistema propuesto pueden ser modificadas, mediante la modificación de las probabilidades de las unidades de activación sin reentrenar el mapa.

Este trabajo se ha realizado en el marco de los proyectos TIN2012-32039, TIN2015-67020-P, y PSI2015-65848-R, financiados por el Ministerio de Economía y Competitividad de España y de los Fondos Feder. Parte del trabajo realizado en esta tesis ha derivado en las publicaciones que se relacionan a continuación.

Principales publicaciones

PCA filtering and Probabilistic SOM for Network Intrusion Detection. Neurocomputing. (Q2). 2015. (Autores: Eduardo De la Hoz Correa, Emiro De la Hoz Franco, Andrés Ortiz García, Julio Ortega Lopera, Beatriz

Prieto). ISSN 0925-2312.
http://www.sciencedirect.com/science/article/pii/S0925231215002982

Lecture Notes in Computer Science: "Network Anomaly Classification by Support Vector Classifiers Ensemble and Non-linear Projection Techniques". Book title: Hybrid Artificial Intelligent Systems. Book subtitle: 8th International Conference, HAIS 2013, Salamanca, Spain, September 11-13, 2013. Proceedings. Springer Berlin Heidelberg. Volume 8073, 2013, pp 103-111. (Autores: Eduardo De la Hoz Correa, Emiro De la Hoz Franco, Andrés Ortiz García y Julio Ortega Lopera). ISBN 978-3-642-40845-8.
http://link.springer.com/chapter/10.1007%2F978-3-642-40846-5_11

Journal Of Theoretical And Applied Information Technology: "Application of FEAST (Feature Selection Toolbox) in IDS (Intrusion Detection Systems)" (SJR: 0,151). (Autores: Eduardo De la Hoz Correa, Alexis Kevin De la Hoz Manotas, Fabio Enrique Mendoza Palechor). ISSN: 1992-8645 ed: v.70 fasc.N/A p.579 - 585 ,2014.

Implementación de GHSOM (Growing Hierarchical Organizing Map) en sistemas de detección de intrusos. Revista de la Facultad de Ingenierías. Universidad de la Costa – CUC. Educosta – Editorial Universitaria de la Costa. Volumen 8. Páginas 117-148. (Autores: Eduardo De la Hoz Correa, Emiro De la Hoz Franco, Andrés Ortiz García y Julio Ortega Lopera). 2012.ISSN 0122-6517.

Participación en otras publicaciones relacionadas

Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps. KBS - Knowledge-Based Systems (Q1). Nederland (Holanda). Volumen 71., 2014, pp 322 - 338. (Autores: Emiro De la Hoz Franco, Eduardo De la Hoz Correa, Andrés Ortiz García, Julio Ortega Lopera y Antonio Martinez Alvarez). ISSN 0950-7051.
http://www.sciencedirect.com/science/article/pii/S0950705114002950

Lecture Notes in Computer Science: "Network Anomaly Detection with Bayesian Self-Organizing Maps". Book title: Advances in Computational Intelligence. Book subtitle: 12th International Work-Conference on Artificial Neural Networks, IWANN 2013, Puerto de la Cruz, Tenerife, Spain, June 12-14, 2013, Proceedings, Part I. Springer Berlin Heidelberg. Volume 7902, 2013, pp 530-537. (Autores: Emiro De la Hoz

Franco, Eduardo De la Hoz Correa, Andrés Ortiz García, Julio Ortega Lopera y Alberto Prieto Espinosa). ISBN 978-3-642-38678-7. http://link.springer.com/chapter/10.1007%2F978-3-642-38679-4_53#page-1.

Modelo de detección de intrusiones en Sistemas de Red, realizando selección de características con FDR y entrenamiento y clasificación con SOM. Revista de la Facultad de Ingenierías. Universidad de la Costa – CUC. Barranquilla-Colombia. Educosta – Editorial Universitaria de la Costa. Volumen 8. Páginas 85-116. (Autores: Emiro De la Hoz Franco, Eduardo De la Hoz Correa, Andrés Ortiz García y Julio Ortega Lopera). 2012. ISSN 0122-6517.

Organización de la memoria

El presente trabajo está estructurado en dos partes, fundamentos y análisis en componentes de conexiones. En el primer capítulo se hace la presentación y los objetivos de la tesis, seguidamente se muestran las principales publicaciones producto del trabajo de investigación realizado, finalizando con la organización de la memoria.

En el capítulo 2, se hace una introducción a la detección de intrusos en redes de computadores, haciendo una descripción de la seguridad informática, enfatizando en lo que realmente debe protegerse cuando se habla de seguridad en sistemas de información. Consecuentemente se definen términos generales de esta temática además de hacer una explicación detallada de las respectivas cualidades que la seguridad debe tener. De igual forma en la Sección 2.2 se aborda el tema de los Sistemas de Seguridad, describiendo en la Sección 2.3 los términos de ataque y amenazas junto con una introducción a los mecanismos de prevención, detección y recuperación en caso de ataques a redes de datos.

La sección 2.3 muestra una visión detallada de los *IDS*, su clasificación y las tendencias a futuro que puedan surgir debido a los nuevos ataques que se presentan hoy día en las redes de datos. El capítulo se inicia haciendo una introducción a los *IDS* abordando definiciones y características, paso seguido en la Sección 2.3.1, se explica cómo se mide la eficiencia de los IDS, explicando los parámetros de medida de los mismos: la precisión, el rendimiento, la completitud, la tolerancia a fallos y el tiempo de respuesta. En la Sección 2.3.2 se muestra como se clasifican los IDS, explicando los basados en fuentes de información y en métodos de detección. El capítulo termina con una muestra de las tendencias futuras de los IDS.
En el Capítulo 3 se ahonda en el aprendizaje estadístico, definiendo la

conceptualización teórica del mismo y los pasos que se utilizan para que a partir de un conjunto de información acerca de un proceso computacional, se pueda construir un modelo que permita predecir nuevos fenómenos asociados a él. Se destaca además la propiedad interesante que tienen estos modelos de emular de manera artificial problemas de ingeniería que requieren para su correcto funcionamiento algún tipo de adaptación al entorno en el que operan. En este capítulo se destacan 5 secciones, las Secciones 3.1 y 3.2 describen el aprendizaje no supervisado y supervisado respectivamente, en la Sección 3.3 se describen los métodos *Kernel*. En la Sección 3.4 se explican los clasificadores estadísticos haciendo énfasis en dos clases: los paramétricos y los no paramétricos o de distribución libre. Para los primeros se enuncia la hipótesis de normalidad en los problemas de clasificación, los clasificadores por mínima distancia y el análisis discriminante de Fisher. Mientras que para los segundos se exponen los términos de estimación de la función de densidad y por criterios de vecindad, así como los clasificadores basados en los criterios de vecindad. En la Sección 3.5 se documenta la selección del subconjunto de características separando los métodos de filtrado de los de envoltura (*wrapper*). A continuación, en la Sección 3.6 es estudian las métricas de rendimiento de un clasificador y las curvas *ROC*. Finalmente, el capítulo termina exponiendo los métodos de validación cruzada por Sub-Muestreo aleatorio repetido, cruzada K-Pliegues y dejar uno afuera (*leave one out*).

En el Capítulo 4 se expondrá el tema de las Redes Neuronales a nivel general, haciendo un recorrido por su arquitectura y forma. Este capítulo no aborda las redes neuronales en su entera profundidad, sin embargo, los límites de esta memoria se concentran en las generalidades de las mismas, su arquitectura, aprendizaje y por último se hace énfasis en las redes autoorganizables. En la Sección 4.1 se hace una generalización de las Redes Neuronales, seguido de las arquitecturas de este tipo de redes expuesto en la Sección 4.2 y los algoritmos de aprendizaje en la Sección 4.3. En la Sección 4.4 se hace una explicación detallada de las Redes Neuronales Autoorganizables, seguido se explican las redes SOM (*Self-Organizing Maps*) donde se detalla su funcionamiento y el algoritmo de entrenamiento.

La segunda parte de esta memoria describe las principales aportaciones de esta tesis. Para ello, el Capítulo 5 muestra un procedimiento innovador basado en mapas auto organizativos probabilísticos, mostrando así el método de selección y clasificación de características propuestos para la detección de anomalías. Este método se ha dividido en tres secciones que resumen el enfoque planteado. La Sección 5.1 Generación de características y filtrado PCA, donde se muestra como se generaron las características para este estudio y se detalla el proceso de filtrado PCA. A continuación, en la

Sección 5.2 titulada Selección de características mediante el Factor discriminante de Fisher para la selección de *eigenvectors* (vectores propios) se da una solución al problema del ruido en el conjunto de datos y de los valores únicos que no son lo suficientemente discriminantes con información redundante o no pertinente y que puede resolverse mediante la selección de los componentes principales de acuerdo con el criterio FDR. Por último, la Sección 5.3 Clasificación usando *Bayesian SOM* detalla el proceso de clasificación usado explicando los mapas auto organizativos Probabilísticos PSOM y la inicialización usada para inicializar el mapa SOM.

En el capítulo 6 se muestra la configuración experimental utilizada. Se describe el conjunto de datos sobre el cual se realizará el estudio de este trabajo de investigación. Además, se detallan otras bases de datos disponibles para la realización de métodos similares como lo es la KDD cup'99. La base de datos se detalla en su totalidad desde sus atributos hasta sus características. Se hace un análisis del procesamiento de los datos, así como del proceso de normalización del conjunto de datos.

En el Capítulo 7, titulado Resultados Experimentales, se muestra un estudio detallado de diferentes métodos de selección y clasificación de características. Se definen los dos experimentos realizados, el primero para determinar el número de características que maximizan el rendimiento de clasificación y el segundo para averiguar el número de unidades de SOM para maximizar el rendimiento.

En el Capítulo 9 Aportaciones y Conclusiones, se resumen las aportaciones a las cuales se llega producto del análisis de los resultados de los experimentos descritos en el capítulo 8.

La memoria concluye con el listado de referencias consultadas.

Parte I

Fundamentos

Capítulo 2. Detección de intrusos en redes de computadores

En este capítulo se hace una introducción a la seguridad informática (sección 2.1), enfatizando en lo que realmente debe protegerse cuando se habla de seguridad en sistemas de información, consecuentemente se definen términos generales de esta temática además de hacer una explicación detallada de las respectivas cualidades que la seguridad debe tener. De igual forma se aborda el tema de los Sistemas de Seguridad (Sección 2.2), describiendo los términos de ataque y amenazas en la Sección 2.3 mencionando algunos de ellos según varios autores, haciendo una introducción a los mecanismos de prevención, detección y recuperación en caso de ataques a redes de datos. Seguidamente, se muestra una visión detallada de los *IDS*, su clasificación y las tendencias a futuro que puedan surgir debido a los nuevos ataques que se presentan hoy día en las redes de datos. Además, se introducen los *IDS* abordando definiciones y características. Paso seguido, se hace una explicación de cómo se mide la eficiencia de los IDS, explicando los parámetros de medida de los mismos: la precisión, el rendimiento, la completitud, la tolerancia a fallos y el tiempo de respuesta. En la sección 2.4.2 se muestra como se clasifican los IDS, explicando los basados en fuentes de información y en métodos de detección. El capítulo termina con una muestra de las tendencias a futuro de los IDS, (sección 2.4.3).

Con el pasar de los años hemos visto cómo las redes de comunicaciones han pasado a ser parte de nuestra vida cotidiana. Las redes de ordenadores han dejado de ser un medio de comunicación para un pequeño grupo de personas, para pasar a ser utilizadas por la mayoría de los ciudadanos y empresas, utilizándose para actividades que varían desde el comercio electrónico hasta el control logístico en puertos y aeropuertos internacionales. Este incremento en el uso de las redes de comunicaciones ha tenido múltiples consecuencias, siendo uno de los más importantes, el aumento de la cantidad y la importancia de la información que fluye por estas redes y la seguridad de la misma.

En el ámbito de la seguridad informática, los intentos de intrusión en redes de computadores han crecido en los últimos años. A diario aparecen programas maliciosos que buscan afectar a un equipo, ya sea para daños locales o para llegar a perjudicar toda una red informática, por lo que es necesario entender los ataques informáticos y la mejor forma de contrarrestarlos, ya sea previniéndolos o detectándolos a tiempo para que su impacto sea menor al esperado por el atacante, siendo estos últimos uno de los retos más importantes a los que se deberá enfrentar la sociedad en los años venideros.

Para contrarrestar este tipo de problemas hoy se dispone de varias herramientas que les dan a las redes un apoyo para mantener la seguridad. Entre estas herramientas encontramos los llamados Sistemas de Detección de Intrusos o IDS. Los IDS basados en red, se encargan de monitorizar el tráfico de una red y se clasifican en el tipo de análisis que llevan a cabo en dos tipos. Los primeros, los IDS de uso indebido, requieren de una base de datos como sistema de apoyo, por lo que necesitan un mantenimiento regular, como son las actualizaciones periódicas. Ejemplos de ellos son los antivirus. Los segundos, son los llamados sistemas basados en anomalías, que no cuentan con ningún soporte, ya que aprenden mediante técnicas estadísticas y/o de inteligencia artificial. Este tipo de algoritmos aprenden a diferenciar entre un comportamiento normal y un comportamiento anormal de los usuarios.

El descubrimiento de nuevos métodos que optimicen los sistemas de detección de intrusiones en las redes de datos, junto con otras metodologías que están siendo implementadas en sistemas de procesamiento automatizado, así como en laboratorios de pruebas tanto en los ámbitos académicos como en la vida práctica están dando una nueva esperanza al tratamiento de la detección de anomalías asociadas a la detección de intrusiones [1]. Existe un claro consenso en la necesidad de desarrollar técnicas de diagnóstico precoz mucho más efectivas, para una rápida intervención, con el único fin de prevenir o disminuir la proliferación de ataques a sistemas informáticos.

La detección a tiempo de las anomalías, ayuda a los administradores de sistemas a crear planes de acción y contingencia a futuro, les da tiempo para discutir las opciones de acción sobre los ataques, mientras no se ha efectuado daño alguno o se está a tiempo de corregir los que sean posibles según la anomalía detectada. Ningún IDS puede detener totalmente la gran cantidad de anomalías que existen hoy día. Sin embargo, detectando a tiempo el tipo de anomalía se puede prevenir en gran medida el efecto de cualquier ataque.

2.1 La Seguridad Informática

Para hablar de seguridad informática primero debemos hablar de la definición de cada uno de los términos que conforman dicha frase. El termino seguridad [2] según la Real Academia de la Lengua-RAE se define como: cualidad de seguro, siendo "seguro" [3] según esta misma entidad el adjetivo que se define como: "libre y exento de todo peligro, daño o riesgo".

Por otro lado, la misma RAE define el termino informática [4] como: "conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores."

Definido los términos que hacen parte de "seguridad informática", pasamos ahora a esclarecer que es lo que se pretende hacer con el conjunto de la definición de ambos términos, *seguridad* e *informática*. Hoy día la seguridad informática lo que pretende escudar es la información [5], siendo esta última la "agregación de datos que tiene un significado más allá de cada uno de éstos", y tendrá un sentido particular según cómo y quién la procese. Un ejemplo muy sencillo sería: 1, 9, 8, 0, son datos, pero si lo vemos como 1980 será una fecha por lo que los datos se convierten en información.

El valor que puede tener la información es relativo y en muchos casos se le da una valoración subjetiva u objetiva debido a su alto grado de intangibilidad, algo que no suele ocurrir con las aplicaciones, los edificios, los equipos de cómputo o la documentación física.

La información puede ser pública o privada, publica como la cantidad de estudiantes de una facultad determinada en cualquier centro educativo, o el porcentaje de alumnos de sexo masculino o femenino del mismo, pero la privada debe ser resguardada como es el caso de las notas finales de cada uno de los estudiantes, por lo que se debe hacer lo posible para preservar dicha confidencialidad, reconociendo características como:

1. La información debe ser ***crítica*** para garantizar la continuidad operativa;
2. ***Valiosa*** ya que es un activo con valor en sí misma y;
3. ***Sensible*** ya que solo debe ser conocida por las personas que tienen acceso a procesarla.

Por la definición de las características anteriores podemos hablar de cualidades como [6]:

- La ***integridad***, característica que hace que la información sea

inalterada a menos que ésta sea modificada por personal autorizado y dicha modificación sea registrada para posteriores controles o auditorias.

- El ***aislamiento*** y la ***confidencialidad*** se relacionan mucho ya que aislar la información se traduce en la capacidad de regular el acceso al sistema impidiendo que personas no autorizadas hagan uso del mismo.
- La ***disponibilidad*** se refiere a la posibilidad de estar siempre útil para aquellos que tienen acceso autorizado a ella, haciendo que la información se mantenga correctamente almacenada con el hardware y el software funcionando en perfecto estado y respetando los formatos para su recuperación.
- La ***autenticidad*** permite decidir si la información es válida y utilizable en tiempo, forma y distribución. Esta cualidad permite también asegurar el origen de la información validando el emisor de la misma.
- ***Protección a la réplica*** es la característica de la información a asegurar que la información solo puede realizarse una vez, a menos que se especifique lo contrario.
- ***No repudio*** es la manera como se verifica que una entidad que recibió o envío la información diga posteriormente que no la envío o recibió.
- ***La consistencia*** es la característica de poder asegurar que el sistema se comporta de la manera como se supone que debe hacerlo ante los usuarios que corresponda, tengan privilegios o no.
- ***La privacidad*** es la característica de la información de ser conocida solo por el personal autorizado para ello. La falta de confidencialidad hace que la información se vuelva obsoleta o provoque daños graves al propietario.
- ***El control*** es la capacidad que se da sobre la información a aquellos usuarios autorizados de decidir cómo y cuándo se permite el acceso a la misma.
- Por último, se encuentra la ***auditoria*** que podemos definir como la capacidad que se tiene sobre el sistema en relación a procesos o acciones y determinar, así como y quién las realiza.

Teniendo en cuenta las definiciones anteriores se enuncia el concepto de *seguridad informática* según [7], ellos la definen como el cumplimiento de confidencialidad, integridad y disponibilidad en un sistema informático.

Al hablar de seguridad informática e información también se debe mencionar las amenazas, siendo estás consideradas como cualquier

elemento que comprometa al entorno del sistema. Las amenazas pueden ser catalogadas en tres espacios de tiempo, antes del ataque, durante y después del ataque. Lo anterior nos lleva a mencionar los mecanismos que garanticen la seguridad de un sistema informático, estos son:

- La ***prevención (antes)***: Son aquellos mecanismos que aumentan la fiabilidad o seguridad del sistema durante su funcionamiento normal.
- La ***detección (durante)***: Aquí se nombran aquellos mecanismos orientados a revelar violaciones de seguridad como lo son los sistemas de detección o prevención de intrusos de los cuales se hablará más adelante.
- La ***recuperación (después)***: Son aquellos mecanismos que se aplican cuando ya el sistema ha sido penetrado para poder llevarlo a su funcionamiento normal.

2.2 Los Sistemas de Seguridad

Los sistemas de seguridad hoy día son muy variados, pero todos cumplen con el precepto de asegurar a toda costa un sistema informático. Para ello el Instituto Nacional para Estándares y Tecnología de los Estados Unidos [8] resumió ciertos estándares con fin de hacer referencia a los requerimientos funcionales mínimos para sistemas operacionales multiusuario:

- ***Identificación y autenticación:*** Es la identificación que cada usuario debe realizar para usar el sistema y cada operación realizada sobre el mismo sistema será registrada con su respectiva identificación.
- ***Reutilización de objetos:*** Es el método por medio del cual se da la posibilidad a múltiples usuarios de acceder a recursos individuales.
- ***Control de acceso:*** Son los derechos y permisos que se le conceden a uno o varios usuarios para acceder a archivos y recursos de red.
- ***Precisión:*** Métodos para proteger los recursos frente a errores, corrupción y accesos tanto autorizados como no autorizados.
- ***Control de cuenta y auditabilidad:*** Son todos aquellos procedimientos que se utilizan para el registro y control de los inicios de sesión de las actividades en los sistemas de red y los enlaces entre ellos, así como las cuentas de los usuarios específicos.
- ***Fiabilidad:*** Es el método que permite asegurar que los sistemas y los recursos estén disponibles y de igual manera protegerlos frente a fallos o perdidas.

- ***Intercambio de datos:*** Es el método para asegurar las trasmisiones de datos con canales de comunicación internos y externos.

Con relación a los datos y sus respectivas características los autores [9] y [10] proporcionan una perspectiva sobre las características que deben ser protegidas y las definen como:

La ***confidencialidad*** es la cualidad que permite que la información de un usuario no pueda ser conocida por otros, la ***integridad*** se refiere a la característica de protección sobre los datos para que otras personas no puedan cambiarlos sin autorización del propietario, y por ultimo está la ***disponibilidad*** que le da al usuario la posibilidad de utilizar sus datos en el momento que él los requiera.

2.3 Ataques y amenazas

Los protocolos de comunicación permiten el método de intercambio de información de dos o más personas. Un ejemplo donde se puede apreciar claramente esta definición es la relación que se establece entre un cliente y un servidor que intercambian información a través de una red. En la actualidad es muy común encontrar también actores maliciosos llamados intrusos que pueden espiar y recoger la información que es transmitida aprovechándose de una vulnerabilidad o hueco de seguridad en los sistemas. Desde que aparece por primera vez el termino *gusano* [11], ha habido una innumerable cantidad de intrusiones de red que se han saltado los mecanismos establecidos para la protección ó n de los sistemas. La masificación de nuevas tecnologías y plataformas han propiciado el esparcimiento acelerado de muchas amenazas que ya existían, y el surgimiento de nuevas gracias a las características propias de la Internet.

Los ataques informáticos son intrusiones ilegales a la seguridad de un sistema, siendo una intrusión la materialización de una amenaza. Una intrusión es definida por [12] como cualquier conjunto de acciones que tratan de comprometer la integridad, confidencialidad o disponibilidad de un recurso. Otras de las definiciones que se utiliza generalmente es la que se proporciona en [13], donde se define la intrusión como un fallo operacional maligno, inducido externamente; aunque es bien sabido que muchas de las intrusiones proceden del interior del sistema de información.

La mayor parte de las intrusiones se realizan a través de los puertos del computador destino, como se explicó anteriormente, haciendo una

exploración de las vulnerabilidades del sistema a atacar. Estos ataques o intrusiones se realizan con programas de escaneo de puertos. Esta técnica de exploración pretende hallar qué servicios ofrece una red o servidor, para realizar conexiones o intentos de conexión a diferentes puertos (TCP o UDP) en la víctima, esperando obtener respuesta de alguno o algunos de ellos, e inferir qué aplicación o servicio está activo en dicho puerto.

Los puertos de un computador pueden encontrarse en varios estados: *abierto*, *cerrado* o *bloqueado*. El estado más vulnerable a un ataque es cuando el puerto se encuentra abierto, esto significa que una aplicación del servidor está escuchando por ese puerto las peticiones de los clientes que se conecten. Por ejemplo, si recibe una respuesta del puerto 22 supondrá que se trata del servicio de SSH, aunque probablemente sea un servidor Web el que esté escuchando peticiones en aquel puerto si el administrador del host así lo ha configurado. Un puerto abierto puede brindar información sobre las vulnerabilidades de seguridad del sistema. Por esta razón, una de las primeras actividades que un atacante intentará realizar en contra de un sistema es sin duda una exploración de puertos por lo que se recomienda tener todos los puertos cerrados a menos que sea estrictamente necesario utilizarlos ya que si se hace efectivo el escaneo de puertos, el atacante obtendrá información básica acerca de los servicios ofrecidos y, adicionalmente, otros detalles del entorno.

Existe una gran cantidad de tipos de escaneo [14], estos pueden basarse en los protocolos TCP o UDP. La diferencia entre ambos radica básicamente en las banderas de protocolo utilizadas tales como SYN, ACK o RST las cuales se aplican sólo a TCP. Entre los tipos de escaneo más comunes se encuentran dos, el TCP Connect y el TCP SYN como se muestra en la Figura 2.1 suministrada por el *Institute for Internet Security* (IIS) [15]. El primero utiliza el proceso de conexión convencional del protocolo TCP conocido como triple *handshaking* o sincronización triple, que intercambia tres mensajes al inicio de una nueva conexión (SYN, SYN_ACK y ACK). El escaneo TCP SYN o semi-abierto no establece una conexión por cada puerto, es más rápido y difícil de detectar.

Es bien conocida la existencia de varios puertos famosos para aprovechar las vulnerabilidades que presentan las aplicaciones que en ellos se ejecutan. Entre ellos están el puerto 22 (ssh) encargado de iniciar sesión, el 23 (telnet), 21 (ftp), el 53 (DNS) encargado de servidores de nombres y otros puertos como el 69 (TFTP) y el puerto 515 (lpd). Teniendo en cuenta lo anterior, a la gran mayoría de usuarios se les recomienda bloquear los puertos que no se usen para asegurarlos, como se mencionó anteriormente.

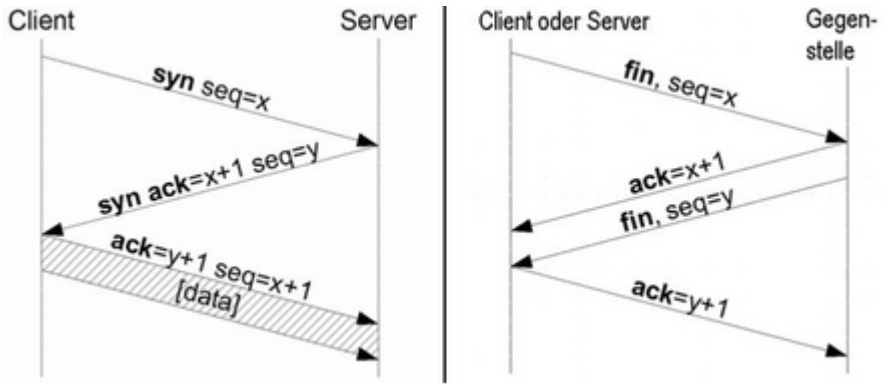


Figura 2.1. TCP Connect y TCP SYN [15].

A continuación, se mencionan algunos de los ataques informáticos más comunes:

- ***Spoofing.*** Este tipo de ataque se caracteriza por la elaboración de tramas TCP/IP utilizando una dirección IP falseada. El objeto de dicho ataque es [16] que el atacante simule la identidad de otra máquina en una red de datos con el propósito de adquirir acceso a recursos de un tercer sistema con el que se ha podido llegar a tener confianza basándose en el nombre o la dirección IP de la maquina suplantada. Este ataque es en la actualidad uno de los más usados por las personas que cuentan con un gran conocimiento del protocolo TCP/IP, ya que muchos sistemas basan su funcionamiento en anillos de confianza, esquema que se basa en hacer que los usuarios de un sistema aporten su clave pública al sistema y firmen las claves del resto de los usuarios, método que no garantiza que una clave es de quien dice ser, más que por la confianza que se pueda tener en el firmante de dicha clave.

- ***Negación de servicio.*** Este ataque es también conocido por sus siglas en inglés como DoS (*Denial of Service*) y en su gran mayoría está dirigido contra un recurso informático, ya sea una máquina o red. En este ataque el o los atacantes tratan de limitar de una manera parcial o total a usuarios legítimos para el acceso a servicios prestados por un recurso informático. Esta característica hace que este ataque se convierta en uno de los ataques más sencillos y contundentes contra todo tipo de servicios planteando en un serio problema, ya que un delincuente informático puede interrumpir constantemente un servicio sin necesidad de grandes conocimientos o recursos,

utilizando programas sencillos o con la ayuda de una gran masa de usuarios ingenuos al ataque. Existen dos tipos principales de ataques de negación de servicio [17]:

- La ***exploración de grietas o desperfectos*** es el ataque que mediante DoS se ayuda de los desperfectos del software para causar una falla en los procesos del sistema y agotar sus recursos. El ejemplo más claro que podemos encontrar de este ataque es el ya reconocido *ping de la muerte* que se encarga de enviar un mensaje tipo ping con unas características anómalas en relación al tamaño en bytes que puede tener normalmente un mensaje de este tipo. Enviando pings con excesos de tamaño se pueden hacer caer servidores o cualquier otro tipo de servicios. Con respecto a los ataques que intentan agotar los recursos del sistema, existen los que afectan el tiempo de procesador, la memoria, el espacio en disco, buffers específicos o ancho de banda de una red.

- Los ataques por ***inundación*** son los otros que podemos encontrar en los tipos de ataques por DoS, y se encargan de inyectar en un sistema o en un componente del sistema más información de la que estos pueden manejar. Con este tipo de ataques no existe ninguna grieta o desperfecto en el sistema que se deba reparar.

Se puede lograr un ataque aún más agresivo si se combinan dos o más técnicas de ataques como es el caso reportado por la revista *ComputerWire* donde se utilizó la técnica de *spoofing* de direcciones IP [16] y un ataque de negación de servicio [18].

- La ***interceptación***, más conocida en el medio de la informática y en el ámbito de la seguridad como *Passive Wiretapping*, es el procedimiento en el cual un agente es capaz de captar información, ya sea cifrada o no cifrada que no iba dirigida a él. Un atacante puede capturar miles de Megabytes de información privilegiada y claves para ser utilizadas más adelante, haciendo que este ataque sea considerado uno de los más peligrosos, ya que no es detectable hasta que se activa.

  La interceptación puede ser implementada utilizando recursos de software como los programas llamados *sniffers*, los cuales se basan en la técnica de "espiar" la red o capturar tramas que circulan por ella de una o todos los computadores conectados a ella. Este ataque también

puede ser realizado por medio de un dispositivo que se conecta directamente a la red pero que en su interior ejecuta un programa de tipo *sniffer*.
Dentro de los ataques de interceptación se pude mencionar otro ataque el cual es comúnmente conocido llamado ***keylogging*** que se caracteriza por registrar las teclas pulsadas por un usuario logrando adquirir contraseñas y otros datos confidenciales.

- Los ***ataques a aplicaciones*** se deben a la utilidad de datos que pueden ser adquiridos en las redes de datos por la estructura física y lógica de las mismas. Como una red se compone de equipos de cómputo aparte de servidores de correo, web, ftp entre otros, los delincuentes informáticos tratan en lo posible de atacar la red en sí y apoderarse del tráfico que es introducido en ellas, más aún si pueden acceder a alguno de los servidores que en ella radican. Entre los ataques más comunes que podemos encontrar a los elementos de una red citamos los ataques al correo electrónico, a los servidores Webs, aquellos ataques que se realizan mediante conexión remota mediante el protocolo SSH [19], diccionarios de datos [20] o programas maliciosos como los troyanos [21] y virus informáticos [22].

La mayoría de los atacantes aprovechan las vulnerabilidades que existen en los sistemas informáticos. Se trata de errores [23] que comprometen la seguridad de un programa o sistema; y que generalmente se debe a la existencia de errores de programación en el código de la aplicación que se ejecuta o puede deberse a errores en la configuración del servicio que se está prestando en ese momento. Es por esto que la seguridad en los sistemas informáticos no solo se le debe promover para los dispositivos adquiridos a elevados precios, sino también afecta a los desarrolladores y a los administradores de software, siendo estos últimos los encargados de "afinar" las aplicaciones para el óptimo desempeño de las mismas con la intención de poder detectar lo antes posible todas las vulnerabilidades posibles que el sistema pueda tener, con miras a corregirlo y evitar ser objeto de ataques potenciales.

El Instituto Nacional de Ciberseguridad de España S.A [24], centro estatal especializado en seguridad en la Red, a través de su Centro de Respuesta a Incidentes de Seguridad de la Información, Inteco-CERT, ha superado la cifra de 20.000 amenazas potenciales analizadas y catalogadas en su base de datos. Esta información es importante para que usuarios, particulares, y empresas conozcan a qué se enfrentan en internet. Cada semana, el Inteco-

CERT localiza y clasifica una media de 21 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.

2.3.1 Mecanismos de prevención

En la actualidad se han desarrollado mecanismos para prevenir o tratar de prevenir ataques informáticos. La finalidad de estos mecanismos es que, en lo posible, dichos ataques no produzcan el daño deseado por los atacantes, o en su defecto no produzcan daño alguno. Tales mecanismos previenen la ocurrencia de violaciones a la seguridad.

Entre los mecanismos más habituales de prevención en las redes de datos de hoy día podemos encontrar [25]:

La ***autenticación*** e ***identificación***, ya que estos hacen posible identificar entidades del sistema de una forma única, y después de ser identificadas, autenticarlas. Se les cataloga como los mecanismos de primera línea de todo sistema informático.
Entre las practicas [26] que se pueden utilizar para la autenticación de usuarios se pueden reconocer: (1) la comúnmente conocida por contraseña que promueve que únicamente el usuario tiene y es su deber y responsabilidad mantener seguro el sistema gracias al par ***usuario-contraseña*** que se le ha asignado; (2) la posesión de un objeto físico que el usuario legítimo posea, como una tarjeta inteligente o credencial que ofrece funciones para un almacenamiento seguro de información y también para su procesamiento; (3) el uso de autenticación biométrica basada en el reconocimiento de alguna característica física de un individuo como sus huellas dactilares o la pupila de sus ojos, y que es una de las técnicas más seguras.
En una segunda posición encontramos los ***mecanismos de control de acceso*** que se han establecido para manipular todos los tipos de acceso sobre un objeto en particular de cualquier entidad del sistema.

Otro de los mecanismos utilizados con el ánimo de separar de la manera más segura posible una maquina o subred de posibles ataques, es el ***cortafuego***. Este se encarga de proteger los servicios y protocolos que desde el exterior puedan suponer una amenaza a la seguridad. En la mayoría de los casos lo que se trata de proteger es el "espacio protegido" llamado comúnmente perímetro de seguridad, el cual brinda "protección" contra una red externa, no confiable, llamada zona de riesgo.
Para garantizar que las comunicaciones de datos por las redes privadas o

públicas sean confiables, se utiliza hoy día la criptografía [27], cifrados de clave pública, privada, firmas digitales, etc. Aunque cada vez se utilizan más los protocolos seguros como SSH o Kerberos [28] como es el caso de sistemas Unix en red. Es un común denominador que también se presente en gran parte de las redes actuales un gran volumen de tráfico sin cifrar, haciendo que los ataques encaminados a robar contraseñas o suplantar la identidad de máquinas de la red se produzcan con más frecuencia.

2.3.2 Mecanismos de Detección

Los mecanismos de detección son los utilizados para detectar violaciones de la seguridad o cualquier tipo de intento que pueda comprometer el óptimo desempeño del sistema. Dentro de los mecanismos de detección se encuentra una herramienta que será explorará más adelante con detenimiento y que hace parte del núcleo base de este proyecto de investigación, hablamos de los *Sistemas de Detección de Intrusos* (o *IDS* por sus siglas en ingles). Estos sistemas son los encargados de supervisar y registrar toda actividad de un sistema para su análisis en busca de alguna actividad maliciosa, y dar así la respuesta más apropiada a dicha actividad. Entre los IDS de disponibles podemos nombrar OSSEC [29], Tripwire [30], Snort [31], RealSecure [32], y Prelude [33], entre otros.

2.3.3 Mecanismos de recuperación

Existen mecanismos que ayudan a recuperar aquellas anomalías que aparecen después de un ataque o violación de un sistema. Con ellos se piensa devolver al sistema informático a su funcionamiento normal. Entre las alternativas que podemos mencionar dentro de estos mecanismos están los antivirus, el uso de hardware adicional, o los mecanismos de detección que incluyan software para la recuperación de sistemas a su estado inicial.

Teniendo en cuenta las tres alternativas mencionadas anteriormente, se puede decir que los sistemas informáticos hoy día están propensos a múltiples ataques y vulnerabilidades posibles ya que la racha de delincuentes informáticos va en aumento y la libertad de información acompañando de políticas de seguridad no planificadas, ni llevadas a la práctica de la manera más eficaz y eficiente posible, permiten a estos delincuentes atentar contra cualquier sistema que no cumpla con los requerimientos mínimos de protección. Aunque ningún sistema es 100% seguro, lo más que podemos decir de él es que es confiable de que se

comporte como se espera hasta que algún ataque no contemplado por los fabricantes o el administrador del sistema luego de su afinamiento logre fragmentar la barrera de seguridad que estos le proveen desde su grado de inteligencia. A continuación, se aborda con detalle el tema de los sistemas de detección de intrusos (IDS).

2.4 Sistemas de Detección de Intrusos - IDS

En la búsqueda de mejorar la complejidad de la auditoría y la habilidad para la vigilancia de sistemas informáticos James P. Anderson [34] en el año 1980 empieza con un trabajo de consultoría realizado para el gobierno de los Estados Unidos introduciendo el término "amenaza" en la seguridad informática y definiéndolo como un intento deliberado de acceso a información, manipulación de la misma, o hacer que un sistema sea inutilizable.

Los IDS supervisan y registran los eventos que ocurren en una computadora o en una red de computadoras. Buscan patrones que permitan identificar intrusiones para responder de la forma más efectiva posible [35], además de evitar malas prácticas, como en el caso de los usuarios autorizados que intentan sobrepasar sus límites de restricción de acceso a la información [36], con el ánimo de poder dar con los responsables del ataque y tomar acciones conducentes a mejorar la vulnerabilidad y castigar, si se puede, a los responsables de dicho ataque.

Además, un IDS dispone de los medios para manejar alertas cuando se detectan signos de intrusión que permitan tomar las medidas correspondientes en el menor tiempo posible. Es por ello que los IDS han ganado terreno en la mayoría de organizaciones que buscan darle un poco más seguridad en sus sistemas informáticos.

Los IDS deberían cumplir las siguientes características:

- Minimizar el consumo de recursos
- Permitir aplicar una configuración según las políticas de seguridad que dicte la organización con el fin de detectar el máximo número de intrusiones.
- Ser capaces de adaptarse a los cambios vertiginosos que sufren los sistemas y los usuarios, además de incluir un proceso rápido y sencillo de actualización.
- Ejecutarse de forma continua, de forma transparente, y con un

mínimo de supervisión

- Tolerar fallos de la red y ser capaces de recuperarse de ellos.
- En caso de verse comprometido alguno de sus componentes, intentar recuperar dicho componente, y en caso contrario administrar un tipo de alerta.

2.4.1 Eficiencia de los Sistemas de Detección de Intrusos

Para evaluar el desempeño de un IDS se han identificado cuatro métricas asociadas a la naturaleza del evento o clase actual, y el estado de la detección o clase predicha. Esas métricas son verdadero positivo (VP - ataque correctamente identificado como ataque), verdadero negativo (VN - tráfico normal correctamente identificado como tráfico normal), falso positivo (FP - tráfico normal identificado incorrectamente como ataque) y falso negativo (FN - ataque identificado incorrectamente como tráfico normal), definidas en [37]. La Tabla 2.1 se denomina matriz de confusión y ha sido definida en [38] y [39]. Entendiendo que los VN y VP corresponden a un funcionamiento correcto de los IDS, en cambio los FN y FP corresponden a un funcionamiento erróneo de los IDS.

Tabla 2.1. Matriz de confusión

Matriz de confusión		**Predicción de Clase**	
		Clase negativa (Tráfico Normal)	Clase Positiva (Ataque)
Clase actual	Clase negativa (Tráfico Normal)	Verdadero Negativo (VN)	Falso Positivo (FP)
	Clase positiva (Ataque)	Falso Negativo (FN)	Verdadero Positivo (VP)

Obviamente, un IDS es más eficiente cuando acierta en mayor medida respecto a la clasificación del tráfico de datos (VN y VP) y presenta baja tasa de fallos en dicha clasificación (FP y FN). Para evaluar formalmente la eficiencia de un IDS es necesario conocer la probabilidad de detectar un ataque (tasa de verdaderos positivos) y de emitir una falsa alarma (tasa de falsos positivos). Estas dos métricas serán definidas a continuación. A partir de estos dos valores se podrá construir la curva ROC (*Receiver Operating Characteristic*) que permitirá valorar el IDS. En la curva ROC de la Figura

2.2, tomada de [39] y conceptualizada en [37] y [40], se aprecia que el IDS perfecto será aquel que detecte todo el tráfico de forma correcta, no generando ninguna falsa alarma.

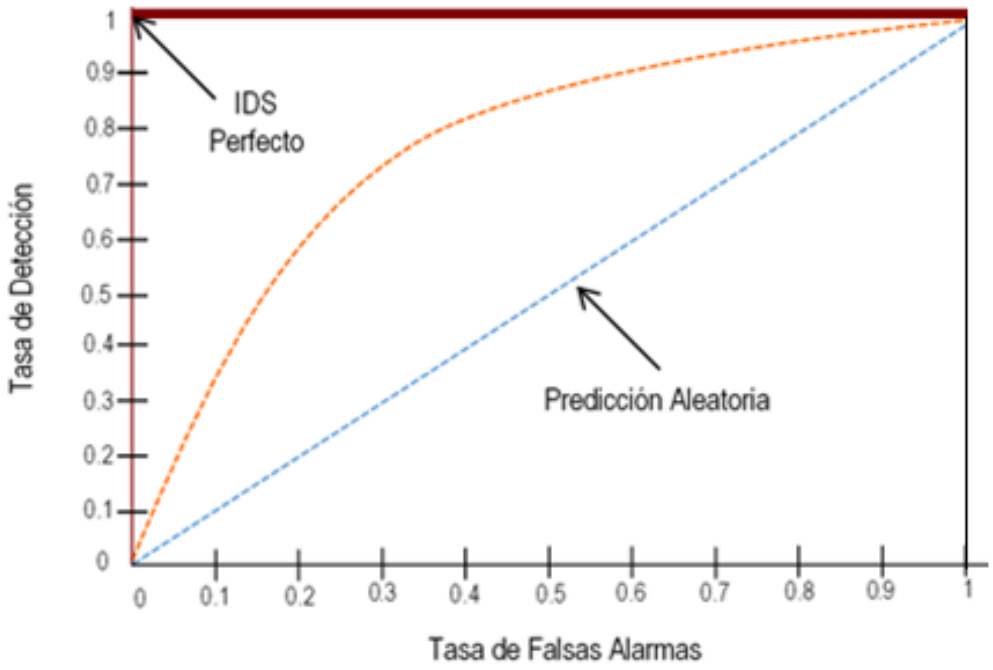


Figura 2.2. Curva ROC

2.4.2 Clasificación de los Sistemas de Detección de Intrusos

Desde las primeras investigaciones realizadas en la década de los 80 por Jame P. Anderson [41] hasta la aparición de sistemas expertos capaces de detectar las desviaciones a partir del comportamiento de diferentes sujetos en el Instituto de Investigación SRI International [42], como se menciona en [43] y [44] entre los años de 1988 y 1990, se han tratado de buscar mejores prácticas que hagan uso de diferentes técnicas y algoritmos para analizar el comportamiento de los sistemas informáticos para darles mayor protección contra ataques de los ya mencionados delincuentes informáticos.

La categorización de los Sistemas de Detección de Intrusos (IDS) ha sido tratada por varios autores, entre ellos podemos mencionar Hervé Debar [45] y Stefan Axelsson [46], así como Wu y Banzhaf [33] quienes los clasificaron según la fuente de datos, el tipo de análisis, y la estrategia de respuesta.

Los IDS por ***fuentes de información*** hacen referencia a la fuente u origen de los datos que se utilizan en el análisis para determinar si una intrusión se ha llevado a cabo. Aquellos que utilizan el ***análisis*** hacen alusión al método de detección utilizado como estrategia. Por último, los que utilizan ***mecanismos de respuesta*** son los que una vez se ha determinado si ha sucedido alguna intrusión pueden, o bien responder de forma activa ante la misma, o bien registrar la detección y no realizar acción alguna.

2.4.2.1 Fuentes de información

Generalmente los IDS requieren manejar una multitud de datos provenientes de distintas fuentes para intentar identificar la presencia de algún tipo de intrusión. El modo en que se recolectan los datos está determinado por la política de recolección de datos, que define el filtro que se usará para la notificación de los eventos. El evento generador produce un conjunto de eventos que representan las fuentes de información del IDS. Se diferencian tres fuentes principales:

- IDS con ***paquetes auditados previamente*** donde los paquetes provienen de un proceso de monitorización previo. Después de un ataque exitoso los datos guardados en los dispositivos de almacenamiento se analizan y se hacen las respectivas correcciones para evitar futuros ataques además de buscar al autor del ataque y el punto de vulnerabilidad que se usó para para acceder al sistema. Esto se llama análisis forense de tráfico de un sistema. La característica principal de estos IDS es que analizan los paquetes de forma mucho más profunda y exhaustiva que otros sistemas. Las respuestas no tienen que hallarse de forma inmediata debido a que el ataque ya ha sucedido y lo importante es recoger todos los datos posibles sobre él. Aunque no se exija inmediatez en las respuestas, sí se valora la rapidez del sistema para detectar el problema de seguridad y así poder cerrar esa vía de acceso evitando futuros ataques.

- Los Sistemas de Detección de Intrusos que se basan en el ***análisis de paquetes recogidos en la red*** son llamados ***sniffers*** y una vez capturado el tráfico en línea, el sistema lo analizará para detectar posibles ataques. La ventaja, como hemos visto antes, es que el tráfico puede procesarse antes de llegar a su destino y, en caso de detectar una intrusión, actuar más rápidamente. El punto negativo de estos IDS es que, si existe un volumen de datos demasiado alto, el coste computacional se incrementará exponencialmente afectando el

rendimiento del IDS.

- Aunque no son muy comunes, también podemos encontrar los IDS de ***análisis del estado del sistema*** que hacen una inspección de los datos que provienen del propio sistema (kernel, servicios, archivos). Estos IDS pueden detectar ciertos comportamientos sospechosos dentro de la propia máquina, ya sea por ser poco frecuentes, o por haber sido previamente identificados como ataques. Por ejemplo, se podría considerar un ataque el uso de unos archivos determinados, o el acceso al sistema de un usuario desde un terminal distinto al habitual. Las ventajas son la detección de ataques internos, y el poco coste computacional del procesamiento de los datos. Como desventaja podemos decir que solo se protege la maquina donde está instalado el sistema, y de la que se analizan los datos.

2.4.2.2 Basados en Detección

Los IDS basados en el método de detección se refieren a la estrategia de análisis que utilizan. Los dos tipos más conocidos de detecciones que un IDS puede realizar son:

- La ***detección de uso indebido (basada en patrones)*** se encarga de comparar patrones de ataque conocidos previamente en busca de coincidencias. Como este tipo de IDS solo detecta las firmas [47], o ataques conocidos, son incapaces de detectar nuevos tipos de ataques, teniéndose que actualizar continuamente su base de datos de firmas, lo que supone una clara desventaja. Un ejemplo de este sistema es el IDS snort [48], que hace detecciones basadas en usos indebidos teniendo como su principal novedad un subsistema flexible de firmas de ataques a partir de una base de datos de ataques conocidos que se actualiza constantemente. Esta base de datos contiene las características de los ataques de red y cuando ocurre uno de ellos, se compara con los patrones almacenados. Si coincide con alguno de ellos se lanza una alerta.

- La detección basada en anomalías detecta ataques a partir de discrepancia, respecto a una regla o un uso [49]. Estos IDS tienen como premisa de que cualquier ataque, o intento, implica un uso anormal de los sistemas [50]. Según lo anterior, un IDS de este tipo comienza por establecer lo que se considera comportamiento normal del sistema (usuarios, redes, registros de auditoría, llamadas del

> sistema de los procesos, etc.). Una vez definido este, clasificará como sospechosas o intrusivas aquellas desviaciones que pueda detectar sobre el comportamiento normal. Por tal motivo, la capacidad de detección dependerá en gran medida de la suposición de que tanto los usuarios como las redes se comportan de un modo suficientemente regular, de forma que cualquier desviación significante pueda ser considerada como evidencia de una intrusión.

La diferencia más clara entre estos dos esquemas se puede observar en imaginar un sistema de detección basado en supervisar las maquinas origen desde las que un usuario sospechoso conecta a un sistema. En un IDS basado en detección de anomalías se tendría una lista de las dos o tres direcciones más utilizadas por el usuario legítimo, alertando al responsable de seguridad en caso de que el usuario conecte desde otro sitio diferente a los habituales. Si utilizáramos un IDS basado en la detección de usos indebidos, la lista que implementaría para detectar los ataques sería mucho más extensa, pero formada por las direcciones desde las que sabemos con una alta probabilidad, que ese usuario no se conectará. De esta forma, si se detectara un acceso desde una de esas máquinas, entonces es cuando el sistema tomarías las acciones oportunas.

Además de estos dos tipos de IDS según como detectan los ataques, también se pueden distinguir dos tipos de IDS. Por un lado, están los que aprenden a partir del comportamiento de los usuarios, de sus procesos, del tráfico de la red, etc., y utilizan métodos estadísticos para su funcionamiento. Por otro lado, están aquellos que especifican el comportamiento del sistema mediante un conjunto de reglas, basándose en determinados parámetros del mismo, pero afrontando el problema a la hora de decidir con precisión qué parámetros delimitan los comportamientos intrusivos.

Consecuentemente con lo expuesto hasta este punto, en los últimos años la aplicación de estrategias de Inteligencia Computacional (*Computacional Intelligence o Soft Computing*) se ha convertido en uno de los campos de investigación más prominente dentro de los IDS como se expresa en [33] y [51]. Sadkhan [52] muestra que este nuevo paradigma se ha venido desarrollando para solucionar problemas que no han podido ser resueltos a través de los métodos tradicionales. Como resultado de estas nuevas técnicas o estrategias en los IDS se han propuesto métodos de detección de anomalías basadas en redes neuronales, conjuntos de lógica difusa, computación evolutiva, sistemas inmunológicos artificiales y sistemas de colonias de hormigas, por señalar algunos. Muchos de estos enfoques son capaces de adquirir e integrar autónomamente conocimiento y pueden ser

implementados a través de un modo de aprendizaje supervisado, o no supervisado.

Las técnicas de detección, desde el punto de vista de la estrategia de análisis, están principalmente basadas en prospecciones propuestas por Noel [53] en [54] y Lazarevic [55], y se muestran en la Figura 2.3.

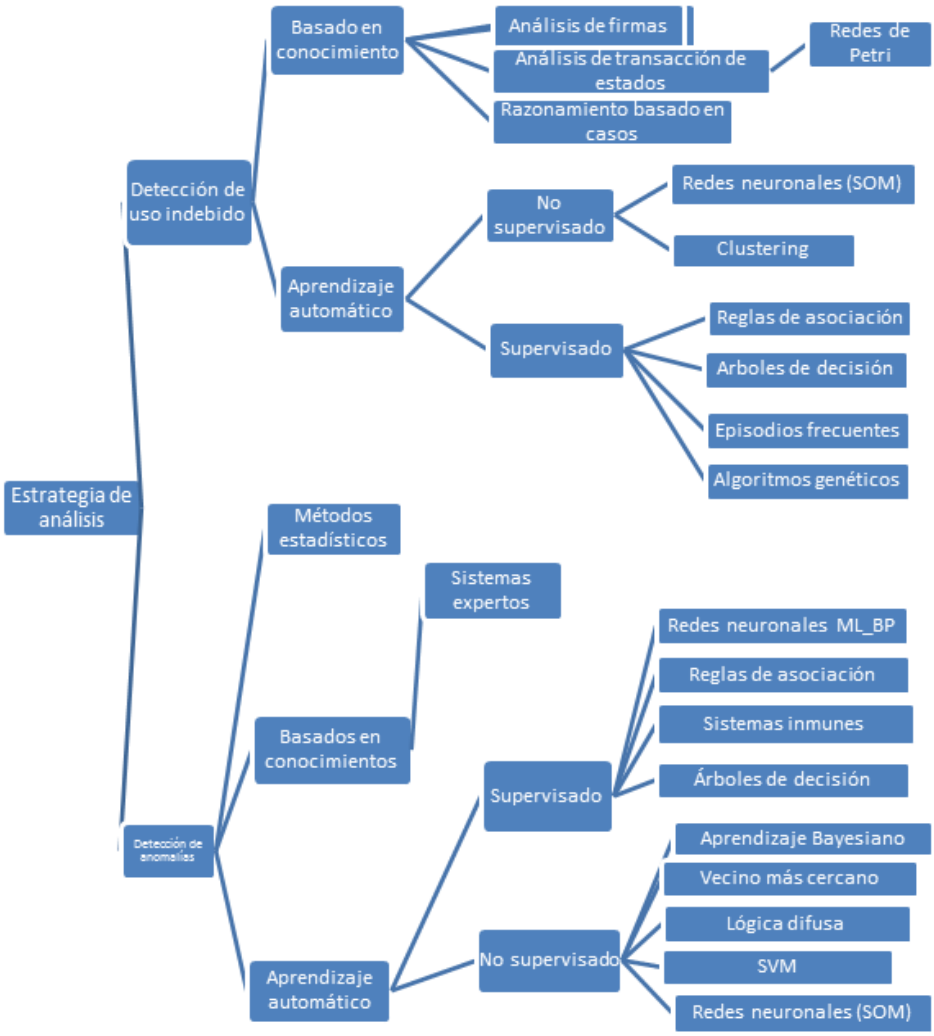


Figura 2.3. Técnicas de detección desde el punto de vista de la estrategia de análisis

2.4.2.3 Respuesta activa o pasiva

La gran mayoría de los IDS cuentan con un método de respuesta básico cuando identifican un ataque. Este método se denomina notificación y es, precisamente, otro de los factores que ayuda a definir el tipo de sistema de detección de intrusos. Así, a partir de los mecanismos de respuesta se pueden diferenciar entre sistemas de respuesta pasiva que, en lugar de tomar

acciones, se limitan a generar la alerta correspondiente, y los sistemas de respuestas activas, que además de generar las alertas correspondientes, reaccionan modificando el entorno.

Los IDS que utilizan las respuestas pasivas [56] generan una alerta cuando detectan un ataque. Por lo general, estas alertas contienen información sobre los paquetes que han hecho saltar una alarma y el motivo por el cual son sospechosos, dejando a los humanos tomar la acción subsecuente basada en esa información. Los IDS con respuestas pasivas tienen una implementación más fácil y un coste computacional bajo. Sin embargo, su gran desventaja es la permanente supervisión de un administrador, que debe analizar las alertas y tomar decisiones, para evitar las intrusiones, o mejorar la seguridad del sistema. Este método de implementación hace necesario evitar los falsos positivos en la mayor medida posible.

Los IDS que proporcionan respuestas activas a los ataques se han implementado en los últimos años dando la posibilidad de responder a los ataques de forma automática. Algunas de las acciones que realizan estos sistemas son cerrar puertos, aislar ciertos terminales, bloquear procesos, etc., dando mejores resultados que los IDS de respuesta pasiva ya que tratan de evitar el ataque, automáticamente, sin que el administrador del sistema tenga que analizar la alarma y actuar en consecuencia como pasa con el otro tipo de sistemas. No obstante, tienen el inconveniente de generar una mayor carga computacional de implementación ya que la complejidad de automatizar la respuesta para el ataque específico que se genere contra el sistema suele utilizar una cantidad de cómputo considerable sobre el hardware que suelen utilizar estos IDS.

Se tienen tres categorías de respuestas activas, la primera de ellas es la posibilidad de incrementar la sensibilidad de las fuentes de información con el fin de capturar la mayor cantidad de datos para poder hacer una comparación exitosa, en tiempo real, de los posibles ataques que pueden ocurrir en el ambiente protegido. La segunda tiene como fin el bloqueo del atacante haciendo una reconfiguración de dispositivos como enrutadores, o sistemas de protección perimetral. La última categoría que encontramos es la de actuar contra el atacante para obtener información de su computador, o del sitio donde se encuentra. Hay que aclarar que la mayoría de ataques se realiza desde direcciones IP falsas o suplantadas.

2.4.3 Tendencias de futuro

Hoy día se trabaja en la implementación de sistemas de detección de intrusos que utiliza métodos avanzados para aumentar la eficiencia y la eficacia en la detección de intrusiones [57-60]: la detección de patrones en ciertas actividades del sistema operativo o en cadenas de tráfico de red relacionadas con potenciales ataques.

Otra de las nuevas tendencias muy utilizadas en la actualidad es, como se ha dicho, la inteligencia artificial, que implica un proceso de aprendizaje para identificar patrones de ataques o actividad anómala. Un ejemplo de este tipo de implementación son las redes neuronales, que se encargan de elaborar modelos dinámicos para los perfiles de uso normal de un sistema. Éstos se van retroalimentando con los diferentes tipos de actividad ya sea anómala o no, para así obtener los perfiles que se adecúen al comportamiento habitual del sistema [58].

Por último, también son prometedores los métodos estadísticos que identifican desviaciones numéricas en ciertos indicadores, basándose en perfiles, previamente definidos, de lo que constituye un ataque o una actividad legítima. Basándose en el análisis estadístico, estos métodos analizan el comportamiento previo de un evento para poder determinar de manera aproximada el que se observaría en futuras repeticiones.

2.5 Conclusiones

Actualmente, las empresas modernas operan y centran gran parte de su actividad a través de la tecnología e Internet. Por tanto, necesitan dotar sus sistemas e infraestructuras informáticas de las políticas y medidas de protección más adecuadas que garanticen el continuo desarrollo y sostenibilidad de sus actividades. En este sentido cobra especial importancia el hecho de que puedan contar con mecanismos y elementos fundamentales de las nuevas tecnologías de seguridad que implementen y gestionen de manera eficaz sus sistemas de información. Como consecuencia, la información, en todas sus formas y estados, se ha convertido en un activo de altísimo valor, el cual se debe proteger y asegurar para garantizar su integridad, confidencialidad, entre otros servicios de seguridad.

La mayoría de los ataques mencionados se basan en fallos de diseño inherentes a Internet (y sus protocolos) y a los sistemas operativos utilizados, por lo que no se podrán resolver en un plazo breve de tiempo. La solución inmediata en cada caso es mantenerse informado sobre todos los tipos de ataques existentes y las actualizaciones que permanentemente

lanzan las empresas desarrolladoras de software, principalmente de sistemas operativos.

En el siguiente capítulo se expondrá con detalle los sistemas de detección de intrusos, haciendo hincapié en la eficiencia de los mismos, su clasificación y sus tendencias a futuro.

En este capítulo se ha iniciado una aproximación al campo de la detección de intrusos. Se trata de un campo muy amplio, en el que existe una gran cantidad de posibilidades sobre las cuales se debe de seguir indagando y avanzando. Entre las posibles líneas de investigación se encuentra la de reducir los falsos positivos que hacen que en muchas ocasiones se generen numerosas cantidades de alertas que no son ataques verdaderos. Así pues, se podría mejorar el motor de detección de los IDS. Otro campo muy interesante es la detección de anomalías, en el cual se debe de seguir avanzando ya aporta un nivel superior de seguridad del que los sistemas de detección de intrusos tradicionales carecen. Su importancia radica en que la detección de anomalías opera sólo desde la base de la actividad normal, buscando eventos extraños al sistema.

En el próximo capítulo se hará una introducción a la teoría del aprendizaje estadístico, mencionando y enfatizando en los métodos y técnicas que se deben utilizar para que este proceso se lleve a cabo.

Capítulo 3. Aprendizaje estadístico

En este capítulo se aborda el tema del aprendizaje estadístico, definiendo la conceptualización teórica del mismo y los pasos que se utilizan para que, a partir de información acerca de un proceso computacional, se pueda construir un modelo que permita predecir nuevos fenómenos asociados a él. Se destaca además la propiedad interesante que tienen estos modelos de emular de manera artificial problemas de ingeniería que requieren, para su correcto funcionamiento, algún tipo de adaptación al entorno en el que operan. En este capítulo se destacan 7 secciones. En las Secciones 3.1 y 3.2 se introducen los términos de aprendizaje no supervisado y supervisado respectivamente. Los métodos *Kernel* se describen en la Sección 3.3, haciendo referencia a las máquinas de soporte vectorial para clasificación y regresión. En la Sección 3.4 se explican los clasificadores estadísticos exponiendo sus dos tipos, los paramétricos y los de distribución libre también llamados no paramétricos. En la Sección 3.5 se documenta la selección del subconjunto de características separando los métodos de filtrado de los de envoltura (o de tipo *wrapper*). Seguidamente, en la Sección 3.6 se estudian las métricas de rendimiento de un clasificador y las curvas ROC (*Receiver Operating Curve*). Finalmente, el capítulo termina exponiendo los métodos de validación cruzada por Sub-Muestreo aleatorio repetido, cruzada K-Pliegues y de dejar uno afuera (*leave-one-out*).

Una rama importante de los métodos que ayudan al reconocimiento de patrones maliciosos en el flujo de datos se basa en las técnicas conocidas como "métodos de clasificación", incluidas en la teoría del aprendizaje estadístico. Estas técnicas consisten en la extracción de cierta información a partir de un conjunto de datos (aprendizaje automático), que se emplea posteriormente para hacer comparativas y diagnósticos con flujos de datos nuevos. En el sentido más amplio, cualquier método que incluye información de un conjunto de muestras para el diseño de una función clasificadora emplea aprendizaje, que en este contexto es un procedimiento algorítmico para reducir el error de la clasificación en el conjunto de entrenamiento. Teniendo en cuenta que la gran mayoría de los problemas de reconocimiento de patrones son complejos se suele invertir la mayor parte del tiempo en la fase de aprendizaje. Un sistema con aprendizaje, puede construir un modelo para predecir nuevos fenómenos. Para ello, el sistema debe tener la capacidad de generalizar, y se hace necesario que el

aprendizaje efectúe algún tipo de inducción a partir de la información disponible.

La inducción en este tipo de aprendizaje precisa de un conjunto de medidas del proceso que se quiere modelar. A este tipo de aprendizaje se le denomina aprendizaje inductivo, y radica en un problema de aproximación de una función de la que se conoce únicamente un conjunto de puntos. La complejidad de este tipo de problemas es notoria ya que el número de variables asociadas al espacio de entrada de la función a aproximar es elevado. Debido a que los procesos modelados por estos sistemas son altamente complejos por su dificultad de caracterización, es habitual que dependan de muchas variables, más aún cuando se trata de entornos reales. Además, las muestras disponibles suelen ser escasas, dispersas y tienen asociada una cierta incertidumbre.

Cuando se extraen muestras de un proceso real, por lo general el costo es alto ya que no se dispone de un número ilimitado de muestras, lo que limita la información disponible para aproximar. De igual forma como el número de variables de entrada en gran medida suele ser elevado, las muestras tienden a estar muy alejadas entre ellas en el espacio de entrada. Lo anterior hace que la correcta reconstrucción de la función se torne más compleja. Además de lo anterior, se debe mencionar también que las muestras por lo general llevan inherentemente una incertidumbre asociada, por lo que no resulta útil aproximar de forma precisa los puntos disponibles. La incertidumbre para este tipo de muestras puede deberse a diversos motivos, entre los que encontramos la imprecisión en la toma de dichas muestras debido a los dispositivos de medida usados, o al ruido en las medidas. Otro punto importante es la información incompleta como puede ser valor de alguna de las variables implicadas. También podemos mencionar un motivo de la incertidumbre el que se trate de un proceso no-determinístico.

Cuando se habla de aprendizaje estadístico, se deben tener en cuenta ciertos componentes, a partir de un conjunto de entrenamiento D de tamaño N y de una función parametrizable $\boldsymbol{F(x;V)}$ (por ejemplo, una red neuronal – Capítulo 4) siendo $\boldsymbol{V}$ el conjunto de parámetros asociados a la función (por ejemplo, los pesos de la red neuronal), el problema del aprendizaje estadístico pasa por calcular $\boldsymbol{V}$ de manera que se logre conseguir un objetivo estadístico, como por ejemplo la minimización de una función de coste estadística.

Existen dos formas de aprendizaje inductivo, el primero es el "**aprendizaje no supervisado**" o "**auto-organizado**", que busca descubrir las propiedades

estadísticas de un vector aleatorio asociado al proceso computacional a modelar. En este caso se dispone de un conjunto de muestras procedentes de variables de entrada del proceso, $D = \{x_i, i = 0 \dots N-1\}$ siendo x_i una muestra aleatoria extraída de un vector aleatorio X de dimensión p que toma valores reales (también pueden ser valores simbólicos codificados como vectores cuyas componentes sean números reales) que opera sobre un espacio de entrada (o escrito de forma abreviada $x_i \in X \in \Re^p$).

El segundo tipo de aprendizaje, denominado "**aprendizaje supervisado**" revela las relaciones existentes entre dos vectores aleatorios, X e Y. Estos dos vectores forman parte del proceso a modelar. Para este aprendizaje se tiene un conjunto de pares de muestras procedentes de dichas variables de entrada y salida de dicho proceso, denominado $D = \{(x_i, y_i), i = 0 \dots N-1\}$ con $x_i \in X \in \Re^p$ e $y_i \in Y \in \Re^m$.

La mezcla del aprendizaje supervisado y el no supervisado se conoce como "**aprendizaje híbrido**". En este caso, el aprendizaje no supervisado facilita el aprendizaje de la parte supervisada puesto que es posible aprender de una forma más fácil las relaciones entre X e Y si previamente tenemos información acerca de la estructura estadística de X.

3.1 Aprendizaje no supervisado

El objetivo final que se desea conseguir con el aprendizaje estadístico no supervisado es la estimación más precisa de la función densidad de probabilidad del vector aleatorio X, denominada $f_X(\boldsymbol{x})$, siendo esta la representación de la estructura estadística de X, que está asociado al proceso computacional que se desea modelar. De esta forma se obtendrá toda la información acerca de X en términos estadísticos.

El aprendizaje no supervisado trata de buscar una formulación más directa del problema mediante la construcción de una función que aproxime al vector aleatorio X. Es decir, que se busca la construcción de una función de $\boldsymbol{x}$ a partir de un conjunto de entrenamiento $D = \{\boldsymbol{x}_i, i = 0 \dots N-1\}$, la cual se denomina $\boldsymbol{F}(\boldsymbol{x}; D)$ que sea un estimador de X tal que:

$$X \approx \hat{X} = \boldsymbol{F}(\boldsymbol{x}; D) \tag{3.1}$$

El criterio más habitual a minimizar durante el aprendizaje se basa en el error cuadrático medio

$$\boldsymbol{F}(\boldsymbol{x}; D) = \frac{1}{2N}\sum_{i=0}^{N-1} \|x_i - \boldsymbol{F}(\boldsymbol{x}_i)\|^2 = \frac{1}{2N}\sum_{i=0}^{N-1}\sum_{j=0}^{p-1}\left(x_{ij} - \boldsymbol{F}_j(\boldsymbol{x}_i)\right)^2 \quad (3.2)$$

Podremos encontrar diversas soluciones al mismo problema estadístico dependiendo de la forma de aproximar $\boldsymbol{F}$. Utilizando un conjunto de vectores prototipos $\{\boldsymbol{w_i}, i = 1 \dots K\}$, $\boldsymbol{F}$ se puede aproximar de forma local de la siguiente forma (a este tipo de sistemas se les conoce con el nombre de cuantificadores vectoriales):

$$\boldsymbol{F}(\boldsymbol{x}) = \textstyle\sum_{i=1}^{K} g_i(\boldsymbol{x})\, \boldsymbol{w}_i \quad \text{con}$$

$$g_i(\boldsymbol{x}) = \begin{cases} 1 \; si \;\; \boldsymbol{x} \in R_i \\ 0 \qquad si \; no \end{cases} \quad \text{y}$$

$$R_i = \left\{\boldsymbol{x} \middle| \|\boldsymbol{x} - \boldsymbol{w}_i\| \leq \min_{j=1\dots K} \|\boldsymbol{x} - \boldsymbol{w}_j\|\right\}$$

Entonces, para minimizar el error cuadrático medio, cada $\boldsymbol{w_i}$ será el centroide que se calcula según las muestras de entrenamiento que caen en la región de Voronoi R_i, lo que resulta en un estimador de la esperanza de pertenecer a

$$\boldsymbol{w}_i = \frac{\sum_{j=0}^{N-1} g_i(\boldsymbol{x}_j)\boldsymbol{x}_j}{\sum_{j=0}^{N-1} g_i(\boldsymbol{x}_j)} \quad (3.3)$$

De manera análoga, al minimizar el error cuadrático medio, los K vectores prototipos se distribuyen sobre el espacio de entrada aproximando de forma discreta la densidad de probabilidad desconocida $\mathrm{f}_X(\boldsymbol{x})$. Esta función de densidad de probabilidad será una buena aproximación usando ese conjunto de vectores prototipo siempre que exista un número de ellos que sea adecuado.

Por el contrario, $\boldsymbol{F}$ puede construir su solución de forma global, para ello se basa en aproximar $\boldsymbol{x} \in \Re^p$ usando un vector $\widehat{\boldsymbol{x}}$ que se obtiene al proyectar

$\boldsymbol{x}$ sobre m ejes determinados donde $m < p$. Esta operación resulta de la transformación lineal a un espacio de dimensión m, así:

$$\boldsymbol{z} = \boldsymbol{V}^T\boldsymbol{x} \text{ con } \boldsymbol{V} = [\boldsymbol{V}_1 \dots . \boldsymbol{V}_m] \tag{3.4}$$

en la ecuación 3.4 los vectores $\{\boldsymbol{V_i}, i = 1 \dots m\}$ tienen módulo 1 y T es la matriz traspuesta. Seguidamente realizando la operación inversa se reconstruye de forma aproximada $\boldsymbol{x}$ así:

$$\boldsymbol{F}(\boldsymbol{x}) = \hat{\boldsymbol{x}} = \boldsymbol{V}\boldsymbol{z} = \boldsymbol{V}\boldsymbol{V}^T\boldsymbol{x} \tag{3.5}$$

Si $\boldsymbol{F}$ minimiza el error cuadrático medio, se puede demostrar que los ejes $\{\boldsymbol{V_i}, i = 1 \dots m\}$ corresponden a las m direcciones, sobre el espacio de entrada donde reside X, en los que las muestras aleatorias $\{\boldsymbol{x_i}, i = 0 \dots N - 1\}$ están más dispersas, o donde los ejes $\{\boldsymbol{V_i}, i = 1 \dots m\}$ estiman aquellas m direcciones de máxima varianza de X. A este tipo de sistemas se les conoce como extractores de componentes principales.

Para el caso presentado, se tomó la esperanza de X como cero en la formulación de la solución de máxima varianza. En caso contrario bastaría substituir $\boldsymbol{x}$ por

$$\boldsymbol{x}' = \boldsymbol{x} - \sum_{i=0}^{N-1} \boldsymbol{x}_i$$

El par de modelos mencionados sirven para distintos fines. Sin embargo, el modelo local tiende a ser más usado en análisis de clústeres por lo que distribuye vectores prototipo sobre X que son centroides de su región de influencia. En este caso si se encuentran cúmulos de datos en X, es factible detectarlos ya que los vectores prototipos contendrán a cada uno de los cúmulos. Por otro lado, el modelo global es implementado más comúnmente como técnica de pre-procesado o reducción de dimensión del espacio de entrada. Lo anterior se debe a que el modelo global construye un espacio de menor dimensión que el espacio de entrada permitiendo comprimir de forma óptima X ya que el modelo, una vez se entrena, es capaz de minimizar el error de reconstrucción que se obtiene de transformar los vectores aleatorios que pertenecen a X, y posteriormente "anti-transformar" volviendo de nuevo al espacio de entrada.

3.2 Aprendizaje supervisado

El principal objetivo del aprendizaje supervisado es descubrir la relación que existe entre dos vectores, X e Y, dados aleatoriamente. Es decir, obtener la relación entre un conjunto de variables procedentes del espacio de entrada y salida del proceso a modelar.

Teniendo en cuenta que se busca caracterizar Y en relación con X, y únicamente disponemos de un número finito de muestras $D = \{(\boldsymbol{x_i}, \boldsymbol{y}_i), i = 0 \dots N-1\}$, se debe utilizar D con el fin de construir una estimación de Y con base de una función del vector aleatorio X, es decir,

$$Y \approx \hat{Y} = \mathbf{F}(\mathbf{x}; D) \tag{3.6}$$

En este tipo de aprendizaje se logran diferenciar dos casos peculiares que hacen de él un caso de estudio aparte, y son la "clasificación" y la "regresión", donde un sistema de clasificación predice una categoría, mientras que una regresión predice un número.

Un ejemplo de ***clasificación*** es el caso del *spam* en los correos electrónicos, donde cada uno se etiqueta como "malicioso o *spam*" o "legítimo" por parte de los usuarios. Otro ejemplo conocido de clasificación es la predicción de bajas en, por ejemplo, un servicio de telefonía. El objetivo en este caso es detectar los patrones de comportamiento de los clientes que predicen que un cliente se clasifica como "baja" o como "no baja".

La ***regresión***, por el contrario, predice un número, como por ejemplo cual va a ser el precio de un artículo, o el número de reservas que se harán en un mes determinado en un hotel.

3.3 Métodos Kernel

Desde que se introdujeron por Vapnik para hacer clasificación, las máquinas de soporte de vectores (*Support Vector Machine*, *SVM*), han emergido multitud de técnicas basadas en el llamado truco del Kernel o *Kernel Trick* [61, 62] y representadas como se observa en la Figura 3.1. Este conjunto de técnicas se les conoce como Métodos Kernel.

Las SVM cuentan con una fundamentación matemática fuerte e ignoran el problema de "*course of dimensionality*" o maldición de la dimensionalidad, con relación a las entradas, escalando así su coste computacional con respecto al

número de datos de entrenamiento. Esta multitud de técnicas llamadas *Kernel Trick* se ha utilizado para obtener extensiones no lineales de otros algoritmos, casos como el Análisis de Componentes Principales (*PCA*) [63], los K-vecinos más próximos [64], u otros algoritmos de clustering [65], ya que se pueden aplicar de manera directa una vez se ha formulado un método en función de los valores de una función sobre pares de entrada en vez de directamente sobre las entradas. De esta forma los métodos que se obtienen son independientes de la estructura concreta de la entrada, por lo que, después de definida la función *kernel*, su complejidad computacional será independiente del número de variables de entrada, ya que fueron definidos en base a valores resultantes de la evaluación de una función sobre pares de datos de entrada.

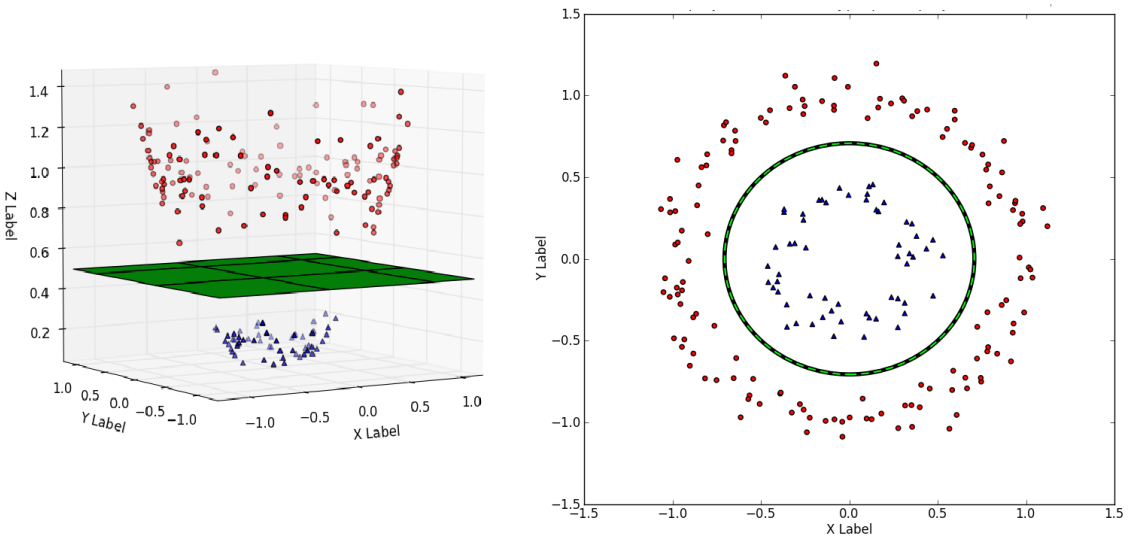


Figura 3.1. Representación del Kernel Trick[1].

3.3.1 *SVM* para clasificación

La gran mayoría de los métodos *kernel* que realizan clasificación se basan en el trabajo de Vapnik de generalizar el clasificador hiperplanar para casos no lineales [62]. Este clasificador binario es una técnica que se concentra en encontrar el "mejor" hiperplano que separe los puntos de dos clases distintas en un espacio *d-dimensional* [66]. El hiperplano considerado el "mejor" define un clasificador que separa los puntos de las dos clases con más generalización, entendiendo que la generalización mayor es la obtenida con el hiperplano que quede a mayor distancia (también llamada "margen de hiperplano separador") de los conjuntos de puntos que separa.

[1] En internet: http://www.eric-kim.net/eric-kim-net/posts/1/imgs/data_2d_to_3d_hyperplane.png

Cabe resaltar que es posible que no exista un hiperplano separador para un conjunto dado de puntos, lo que da la posibilidad de flexibilizar la condición introduciendo variables de holgura y parámetros adicionales para controlar la proporción de error/precisión y así permitir que se clasifiquen mal algunas muestras.

La complejidad computacional de SVM para clasificación depende del número de puntos de entrenamiento, lo que dificulta su aplicación a grandes conjuntos de datos. No obstante, aunque en la literatura existen aproximaciones para aliviar este problema como es el caso del uso del procesamiento paralelo [67].

3.3.2 SVM para regresión

Este tipo de SVM es una modificación del modelo para clasificación. En este caso, dado un conjunto de muestras de una función $\{(x_1, y_1), \dots, (x_N, y_N)\} \subset X \times \Re$, donde X es un conjunto donde existe definida una operación de producto escalar entre sus elementos. La técnica de la regresión $\varepsilon -$ SVR busca una función $f(x)$ tan suave como sea posible cuyos valores difieran en cada salida real y_1 no más de un valor ε para todo $x_i \in X$. Esta función también se denomina función de error $\varepsilon - insensible$.

A pesar de que las SVM para regresión tienen serios inconvenientes frente a las SVM de clasificación, este modelo de clasificación ha sido empleado con éxito en algunos trabajos para modelado de series temporales, como es el caso de la predicción de cargas eléctricas [68], o series financieras [69].

3.4 Clasificadores estadísticos

En la presente sección se describe el concepto de la clasificación, enmarcado en los métodos de clasificación y los diferentes clasificadores estadísticos con el fin de que, en las próximas secciones de los capítulos de esta memoria, se puedan relacionar entre sí con los demás conceptos utilizados.

Cuando se aborda el termino *clasificación* se presenta en algunos casos alguna confusión, ya que al momento de clasificar objetos, individuos o cualquier otro tipo de elementos se pueden presentar problemas de diferentes tipos. Debido a esta problemática en particular se han planteado tres tipos de problemas bajo una misma denominación común de clasificación.

El primero de ellos corresponde a la *discriminación*, este supone la existencia de dos o más poblaciones de las que se ha logrado extraer muestras de los individuos. Consecuentemente el problema planteado es la capacidad de obtener reglas que permitan la asociación, de un individuo nuevo a alguna de las poblaciones existentes, sin conocer su población origen. Un segundo tipo es el de *agrupamiento* (*clustering*). Para este caso concreto se cuenta con una muestra de individuos sin conocer a priori las poblaciones a las que pertenecen, siendo la problemática clasificatoria la de enmarcar los individuos en grupos lo más diferenciados posibles. Por último, encontramos a los problemas de *disección*, estos buscan dividir en grupos una muestra o población dada, así sean muestras homogéneas, estableciendo las respectivas limitaciones, sean o no naturales.

Al momento de implementar el proceso de la clasificación, se debe dejar claro ciertos conceptos. El primero de ellos es el "*patrón*", el cual se define como una observación modelo de una variable aleatoria que caracteriza una clase. Es decir, se define espacio de observaciones, espacio experimental o espacio de representación asociado a un vector aleatorio $\boldsymbol{X}$ y denotamos Φ al conjunto de todos los posibles valores que puede tomar un patrón $\boldsymbol{x}$, así,

$$\boldsymbol{\Phi} = \prod_{i=1}^{n} \boldsymbol{\Omega}_i \qquad (3.7)$$

donde $\boldsymbol{\Omega}_i$ representa el espacio muestral de la variable X_i **y** $\boldsymbol{\Pi}$ denota el producto cartesiano.

Las clases (grupos o categorías) que se consideren útiles establecer en las observaciones experimentales $\{\boldsymbol{x}\}$, están representadas por $\{C_1, C_2, \dots, C_k\}$. Puede ocurrir que $\boldsymbol{x}$ no pueda asignarse a ningún grupo o categoría, debido a que su *clase original* no haya sido prevista con anterioridad, o debido a la tenencia de valores "anómalos". Este tipo especial de observaciones que no se logran clasificar, se incluyen en una clase especial denominada clase de rechazo y denotada por C_0 de esta forma el conjunto de clases quedara como:

$$\boldsymbol{\Phi} = C_0 \cup C_1 \cup C_2 \cup \dots \cup C_k \qquad C_i \cup C_j = 0, \quad \forall i \neq j$$

Consecuentemente una variable supervisora será una función $t: \Phi \rightarrow \{t_1, t_2, \dots, t_k\}$ que asocia cada observación $\boldsymbol{x} \in \Phi$ con una etiqueta de clase. Esta clase que es representada por la etiqueta, se denomina *clase cierta* o *segura.*

3.4.1 Concepto de clasificador

Un clasificador o regla de clasificación es una aplicación $\psi: \Phi \rightarrow \{t_0, t_1, \dots, t_k\}$ que asocia a cada observación $\boldsymbol{x}$, una etiqueta de clase $t^{\boldsymbol{x}} \in \{t_0, t_1, \dots, t_k\}$ que indica la clase a la que debe pertenecer.

Teniendo en cuenta la definición anterior, una clase discriminable estará dada por la representación de observaciones de diferentes clases, obteniendo generalmente una distinción entre sí de los grupos presentes, dando la posibilidad de asociar regiones disjuntas en el espacio de las observaciones a cada una de las clases representadas. Así bien el objetivo de un clasificador, consiste en dividir el espacio de observaciones Φ en regiones disjuntas, $R_0, R_1, \dots, R_k$, mediante una regla que permita asociar cada observación $\boldsymbol{x}$ a una clase C_i si y solo si $\boldsymbol{x} \in R_i$. La obtención de los R_i puede considerarse como el cálculo de la función inversa ψ.

Los clasificadores se dividen en dos grandes tipos, según el criterio de clasificación y el criterio basado en el conjunto de la distribución de la probabilidad. Los primeros se basan en la estrategia de construcción del clasificador concepto que se fundamenta en el método de clasificación, mientras que los segundos hacen uso de los métodos paramétricos y no paramétricos.

3.4.2 Muestras de aprendizaje y de validación

Siendo $\mathcal{M}$ una muestra aleatoria $\mathcal{M} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ donde cada observación $\boldsymbol{x}_i \in \Re^n$, una *muestra de aprendizaje* o de *entrenamiento* es el conjunto de patrones $\wp \in \mathcal{M}$ utilizado para la obtención del modelo $y(\boldsymbol{x}, \boldsymbol{w})$. Mientras que una *muestra de validación* o de *prueba* es el conjunto de patrones $\mathcal{B} \in \mathcal{M}$ reservados para validar el modelo, es decir, para estimar su capacidad de generalización.

En base a lo anterior, existen dos tipos de error asociados a la bondad de $y(\boldsymbol{x}, \boldsymbol{w})$. El primero es la *función de pérdida* o *función criterio* utilizada para entrenar el modelo, es decir, para obtener sus parámetros denotada por $err(y(\boldsymbol{x}, \boldsymbol{w}), \wp)$, y el *error de generalización* o *de validación* utilizado para evaluar el comportamiento del modelo sobre nuevas observaciones, denotado por $err(y(\boldsymbol{x}, \boldsymbol{w}), \mathcal{B})$.

Cuando se habla de la construcción de un clasificador, se debe elegir con

detenimiento el modelo a usar, ya que es en este punto donde aparecen diferentes puntos a tener en cuenta como son: la robustez, la convergencia del mismo en un tiempo prudencial, su dependencia del número de patrones de entrenamiento, la dimensión del espacio de características, su capacidad de generalización o simplificación, etc. Serán estos índices los que favorecerán o empeorarán las soluciones sencillas frente a las complejas.
A continuación, se describen dos clasificadores lineales sencillos que dan una idea de los conceptos introducidos hasta el momento. Ambos definen un hiperplano en el espacio de características.

Los clasificadores lineales actuando sobre $x \in \Re^m$ se pueden expresar como:

$$f_{lineal}(x) = x \cdot \omega + \omega_0 \qquad (3.8)$$

donde el vector $\boldsymbol{\omega}$ y el escalar ω_0 forman un conjunto de $m + 1$ parámetros que han de ser estimados. Se muestran a continuación dos de los métodos utilizados para su estimación.

3.4.3 Clasificadores paramétricos

En esta sección se describen las técnicas de reconocimiento estadístico de patrones, desde un punto de vista paramétrico. La correcta clasificación de los patrones basándose en el conocimiento parcial de las funciones de densidad de los patrones de cada clase, es como tal el objetivo primordial de las reglas de clasificación paramétrica. Por lo general se suele exigir que dadas las clases las distribuciones de probabilidad de las observaciones sean normales multivariantes de parámetros conocidos.

Distancia de Mahalanobis y distribución normal. Formalizada por el matemático hindú Mahalanobis en 1930, es una de las formulaciones de distancia que mayor interés tiene, con miras a la clasificación, para definir una métrica basada en la dispersión de los datos usando la matriz de covarianza $\sum$.

Se define así, sea $\boldsymbol{X} \rightarrow \mathcal{N}(\boldsymbol{\vartheta}, \sum)$, donde $\boldsymbol{X}$ es una variable aleatoria normal n-multivariante de vector de medias $\boldsymbol{\vartheta}$ y $\sum$ la matriz de covarianza. Por construcción, $\sum$ es simétrica y semi-definida positiva. Si $\sum$ es invertible, se presenta un producto escalar tal que $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x} \sum^{-1} \boldsymbol{y}$ cuya norma de vector asociada $\|\boldsymbol{x}\|_{\sum} = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$ conduce a una métrica que tiene en cuenta la variabilidad de las componentes de $\boldsymbol{X}$ en las distintas dimensiones del

espacio.

Entonces, si $\boldsymbol{x}$ e $\boldsymbol{y}$ son dos observaciones de una variable $\mathcal{N}(\boldsymbol{\vartheta}, \sum)$, la distancia de Mahalanobis entre $\boldsymbol{x}$ e $\boldsymbol{y}$ se define como:

$$d_{\sum}(x, y) = (x - y)^T \sum^{-1} (x - y) \tag{3.9}$$

3.4.3.1 Clasificadores por mínima distancia

La distancia Mahalanobis se define como una métrica capaz de considerar la variabilidad de las componentes de $\boldsymbol{X}$ en las distintas direcciones espaciales. Por ende, basándose en esta métrica, es posible construir un clasificador como casos particulares.

Clasificación por mínima distancia de Mahalanobis. Si las matrices de covarianza son iguales, $\sum_i = \sum \quad, \forall_i = 1, \ldots, p$, existirán funciones discriminantes que se pueden simplificar como:

$$h_i(\boldsymbol{x}) = \boldsymbol{x}^T \sum^{-1} \boldsymbol{\vartheta}_i - \frac{1}{2} \boldsymbol{\vartheta}_i^T \sum^{-1} \boldsymbol{\vartheta}_i, \tag{3.10}$$

dando como resultado el clasificador lineal:

$$\psi(x) = t_i \Leftrightarrow i = arg \max_{j=1,\ldots,p} \left\{ x^T \sum^{-1} \boldsymbol{\vartheta}_j - \frac{1}{2} \boldsymbol{\vartheta}_j^T \sum^{-1} \boldsymbol{\vartheta}_j \right\} \tag{3.11}$$

conocido en la literatura como el *clasificador por mínima distancia de Mahalanobis*. Este clasificador cuenta con una regla de clasificación de mínima distancia de Mahalanobis, dada por:

$$\psi(\boldsymbol{x}) = t_i \quad \Leftrightarrow \quad D_i(\boldsymbol{x}) < D_j(\boldsymbol{x}) \ \forall\, j = 1, \ldots., p \tag{3.12}$$

donde $D_i(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{\vartheta}_i)^T \sum^{-1} (\boldsymbol{x} - \boldsymbol{\vartheta}_i)$.

Clasificación por mínima distancia euclídea. Uno de los casos particulares del clasificador por mínima distancia de Mahalanobis se da cuando $\sum_i = \sigma^2 I$. Geométricamente, las clases son hiperesferas de centros $\boldsymbol{\vartheta_i}, i = 1, \ldots, p$ e igual tamaño, ya que al estar las X_i, escaladas por igual en todas las direcciones del espacio la métrica asociada es equivalente a la euclídea. En

este caso $D_i(x) = (x - \vartheta_i)^T (x - \vartheta_i)$ coincide por tanto con la distancia euclídea (al cuadrado) de $\boldsymbol{x}$ a la clase C_i.

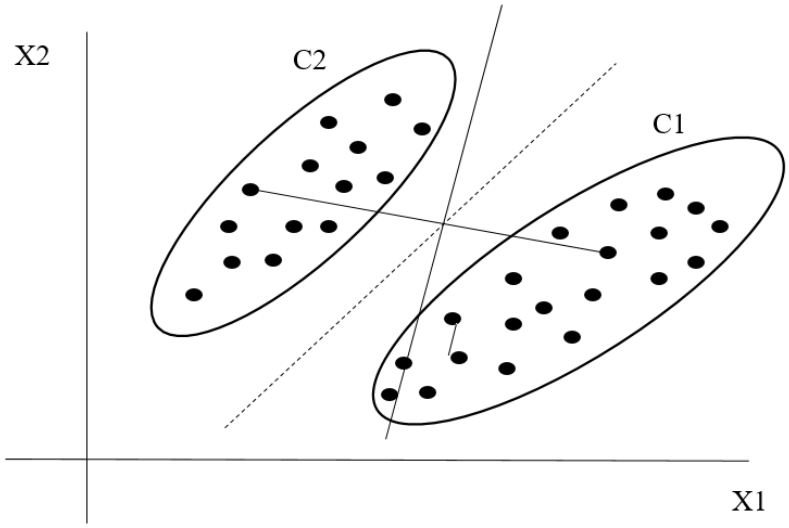


Figura 3.2. Clases linealmente separables, mal clasificadas por el clasificador de mínima distancia euclídea y bien clasificados por el clasificador Mahalanobis

Ahora bien, un clasificador por mínima distancia euclídea se define como:

$$\psi(x) = t_i \quad \Leftrightarrow \quad \| x - \vartheta_i \|^2 < \| x - \vartheta_j \|^2 \quad \forall j = 1, \dots, p \qquad (3.13)$$

El clasificador de mínima distancia cuando se ejecuta, funciona de manera correcta cuando la distancia entre las medias es grande en comparación con la media de dispersión de cada clase, o cuando las dispersiones de las distintas clases son aproximadamente iguales. En la Figura 3.2 se hace una comparativa entre los clasificadores por mínima distancia euclídea y de Mahalanobis. Para el caso euclídeo la frontera discriminante es perpendicular al segmento que une los centros de las clases, esto hace que la clasificación sea incorrecta. Cuando se aplica clasificación por Mahalanobis, el proceso adapta la frontera discriminante para conseguir la mejor separación.

Clase de rechazo. Para cualquiera de los casos, ya sea el de mínima distancia de Mahalanobis o el de mínima distancia euclídea, el valor umbral se calcula como el número de desviaciones típicas al centro de la clase. La regla de clasificación de (3.12) y (3.13) queda como sigue:

Sea $D_\Sigma(x)$ la mínima distancia de Mahalanobis de x a los centroides $\boldsymbol{\vartheta}_i$ y θ_i, el valor umbral de distancia prefijado para la clase C_i. La regla de mínima

distancia de Mahalanobis con clase de rechazo se escribe como sigue:

$$\psi(x) = \begin{cases} t_i & si\ D_{\sum i}(x) < \theta_i \\ t_0 & si\ D_{\sum i}(x) \geq \theta_i \end{cases} \tag{3.14}$$

3.4.1.3 Discriminante lineal de Fisher

El discriminante lineal de Fisher es una particularización del Análisis Discriminante Lineal (ADL), siendo este último un método utilizado en estadística, reconocimiento de patrones y aprendizaje de máquinas para encontrar una combinación lineal de rasgos que caracterizan o separan dos o más clases de objetos o eventos.

La idea de la combinación lineal de variables es la base del trabajo de Fisher para la discriminación de grupos, presentando las siguientes características:

- El paralelismo entre la función criterio del Análisis Discriminante Lineal (ADL) y la de los clasificadores neuronales.
- Cuando se trata de problemas de clasificación biclase, la coincidencia de los resultados utilizando el ADL y redes neuronales.
- El planteamiento de modelos neuronales que trabajen sobre espacios de dimensión reducida gracias a la reducción de dimensión que conlleva el ADL.

El análisis discriminante lineal se centra en encontrar las combinaciones lineales de variables tales que las nuevas variables, tengan una gran varianza entre los grupos, buscando así resaltar las diferencias, y una varianza interna pequeña, es decir, que los elementos estén lo más cohesionados posibles.

El ADL parte de una estructura de datos formada por p grupos y N observaciones de dimensión n, exigiéndose en este caso que las variables discriminantes tengan una escala de intervalo o razón. Expresándose de manera matricial se tendría:

$$\begin{bmatrix} \mathrm{E}_1 \\ \mathrm{E}_2 \\ \vdots \\ \mathrm{E}_r \end{bmatrix} = \begin{bmatrix} \nu_{01} \\ \nu_{02} \\ \vdots \\ \nu_{0r} \end{bmatrix} = \begin{bmatrix} \nu_{11} & \nu_{12} & \cdots & \nu_{1n} \\ \nu_{21} & \nu_{22} & & \nu_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \nu_{r1} & \nu_{r2} & \cdots & \nu_{rn} \end{bmatrix} \begin{bmatrix} \mathrm{H}_1 \\ \mathrm{H}_2 \\ \vdots \\ \mathrm{H}_n \end{bmatrix} \Leftrightarrow \boldsymbol{E} = \boldsymbol{\nu}_0 + \boldsymbol{V}_{r\times n}^{T}\, \boldsymbol{H}$$

Observándose que:

- Cada E_i es una variable aleatoria obtenida como combinación lineal de las variables discriminantes y representa una función discriminante o eje discriminante.
- Geométricamente, $\mathrm{E}_i(\boldsymbol{x})$ es la proyección de x sobre el eje de vector director $\boldsymbol{\nu}_i$.
- Los parámetros ν_{0i} y $\boldsymbol{\nu}_i^T = [\,\nu_{i1}, \nu_{i2}, \ldots, \nu_{in}]$ representan el sesgo y los pesos de la función E_i. Se usa ν_{ij} en vez de w_{ij} para representar los pesos según la notación estadística clásica, y w_{ij} representarán los elementos de la matriz suma de cuadrados y productos cruzados intragrupo. Si las variables H_i están ya tipificadas, pueden eliminarse los sesgos.
- Las estimaciones de las medias y de las varianzas muestrables de E_i sobre las clases C_j pueden obtenerse como:

$$\overline{E_{ij}} = \nu_{i0} + \frac{1}{N_j}\sum_{x\in C_j} \boldsymbol{\nu}_i^T \boldsymbol{x} = \nu_{i0} + \boldsymbol{\nu}_i^T \widehat{\boldsymbol{\vartheta}}_j,\ \ S_{ij}^2 = \frac{1}{N_j - 1}\sum_{x\in C_j} (\boldsymbol{\nu}_i^T x - \boldsymbol{\nu}_i^T \widehat{\boldsymbol{\vartheta}}_j)^2 \qquad (3.15)$$

Haciendo la descomposición de la variable total, se denomina la matriz de *scatter* entre-clases o matriz de suma de cuadrados y productos totales, denotada por $\boldsymbol{P}_{n\times n}$, a la matriz definida por:

$$\boldsymbol{P}_{n\times n} = \sum_{x} \left(\boldsymbol{x} - \widehat{\boldsymbol{\vartheta}}\right)\left(\boldsymbol{x} - \widehat{\boldsymbol{\vartheta}}\right)^T \qquad (3.16)$$

esta matriz contiene la información de la variabilidad total que presenta el conjunto de n variables independientes. De igual forma, la matriz de *scatter* dentro de una clase o matriz de suma de cuadrados y productos cruzados intragrupos, denotada por $\boldsymbol{G}_{n\times n}$, se define por:

$$\boldsymbol{G}_{n\times n} = \sum_{i=1}^{p}\sum_{x\in C_i} \left(\boldsymbol{x} - \widehat{\boldsymbol{\vartheta}}_i\right)(\boldsymbol{x} - \boldsymbol{\vartheta}_i)^T \qquad (3.17)$$

Consecuentemente, se denomina matriz de suma de cuadrados y productos cruzados intergrupos y se denota por $\boldsymbol{D}_{n\times n}$ a la matriz definida por:

$$\boldsymbol{D}_{n\times n} = \sum_{i=1}^{p} N_k \left(\widehat{\boldsymbol{\vartheta}}_i - \widehat{\boldsymbol{\vartheta}}\right)\left(\widehat{\boldsymbol{\vartheta}}_i - \widehat{\boldsymbol{\vartheta}}\right)^T \qquad (3.18)$$

Se pueden concluir entonces dos cosas. La primera, es que las matrices $\boldsymbol{G}$ y $\boldsymbol{P}$ tienen similaridad y lo que las diferencia es que $\boldsymbol{G}$ calcula las desviaciones de cada observación respecto de la media se su grupo. Segundo, que las matrices $\boldsymbol{G}$ y $\boldsymbol{D}$ están contenidas por información básica de las dispersiones dentro de cada categoría y entre las categorías respectivamente. Ahora bien, $\boldsymbol{T} = \boldsymbol{G} + \boldsymbol{D}$, puede calcularse $\boldsymbol{D}$ como $\boldsymbol{T} - \boldsymbol{G}$ ya que la variabilidad total puede descomponerse en la suma de la variabilidad intragrupo y la variabilidad intergrupo.

Para el caso de **dos categorías** (*clasificación biclase*), es suficiente con una sola función discriminante ($r=1$) como clasificador. La representación más común del eje discriminante con relación a los grupos se observa en la Figura 3.3. Se busca un vector de pesos ϑ que alcance la máxima capacidad de clasificación, tal que haga máxima la distancia entre los centroides minimizando a su vez las dispersiones dentro de los grupos. Por tanto, se busca la solución de:

$$\max_{\boldsymbol{v}} J(\boldsymbol{v}) \text{ donde } J(\boldsymbol{v}) = \frac{\left|\boldsymbol{v}^T\widehat{\boldsymbol{\vartheta}}_1 - \boldsymbol{v}^T\widehat{\boldsymbol{\vartheta}}_2\right|^2}{Q_1^2 + Q_2^2} \qquad (3.19)$$

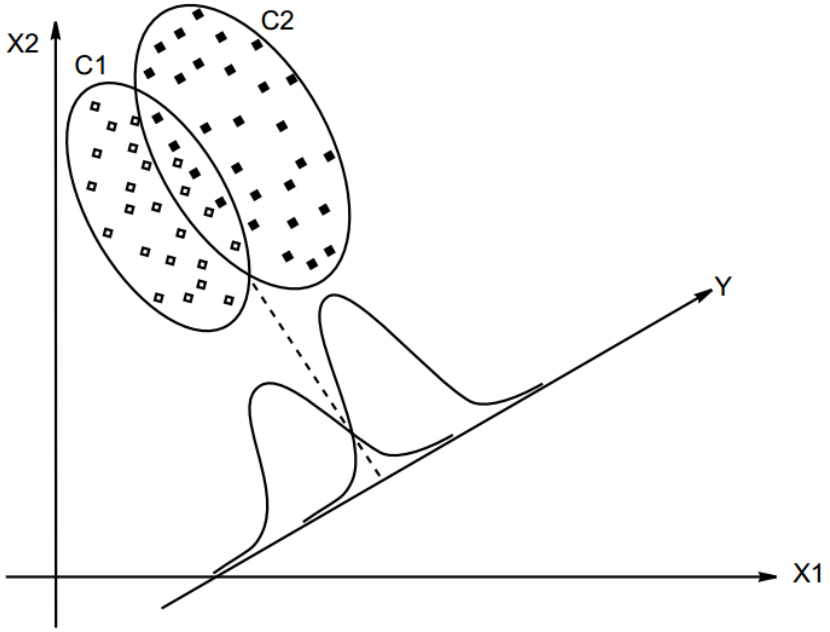


Figura 3.3. Representación del ADL para dos clases

Para el caso de dos grupos, las matrices sumas de cuadrados intergrupo e intragrupo estarán representadas así:

$\text{SCI}_{\boldsymbol{er}}$: $\boldsymbol{D}_{n\times n} = \left(\widehat{\boldsymbol{\vartheta}}_2 - \widehat{\boldsymbol{\vartheta}}_1\right)\left(\widehat{\boldsymbol{\vartheta}}_2 - \widehat{\boldsymbol{\vartheta}}_1\right)^T$

$\text{SCI}_{\boldsymbol{ra}}$: $\boldsymbol{G}_{n\times n} = \sum_{x\in C_1}\left(\boldsymbol{x} - \widehat{\boldsymbol{\vartheta}}_1\right)\left(\boldsymbol{x} - \widehat{\boldsymbol{\vartheta}}_1\right)^T + \sum_{x\in C_2}\left(\boldsymbol{x} - \widehat{\boldsymbol{\vartheta}}_1\right)\left(\boldsymbol{x} - \widehat{\boldsymbol{\vartheta}}_2\right)^T$

Ahora bien, como:

$$Q_1^2 + Q_2^2 = \boldsymbol{v}^T\boldsymbol{G}\boldsymbol{v}, \quad \left|\boldsymbol{v}^T\widehat{\boldsymbol{\vartheta}}_1 - \boldsymbol{v}^T\widehat{\boldsymbol{\vartheta}}_2\right|^2 = \boldsymbol{v}^T\left(\widehat{\boldsymbol{\vartheta}}_2 - \widehat{\boldsymbol{\vartheta}}_1\right)\left(\widehat{\boldsymbol{\vartheta}}_2 - \widehat{\boldsymbol{\vartheta}}_1\right)^T\boldsymbol{v} = \boldsymbol{v}^T\boldsymbol{D}\boldsymbol{v}$$

da como resultado que $\boldsymbol{J}(\boldsymbol{v})$ pueda expresarse como el cociente entre $\text{SCI}_{\boldsymbol{er}}$ y $\text{SCI}_{\boldsymbol{ra}}$ así:

$$\boldsymbol{J}(\boldsymbol{v}) = \frac{\text{SCI}_{\boldsymbol{er}}}{\text{SCI}_{\boldsymbol{ra}}} = \frac{\boldsymbol{v}^T\boldsymbol{D}\boldsymbol{v}}{\boldsymbol{v}^T\boldsymbol{G}\boldsymbol{v}} \tag{3.20}$$

representando un problema de autovalores generalizados según Schott [70].

Cuando se presentan más de dos categorías (clasificador multiclase), las sumas de cuadrados intergrupos e intragrupo para E pueden obtenerse conociendo las de H haciendo uso de la expresión $\boldsymbol{E} = \boldsymbol{v}_0 + \boldsymbol{U}^T\boldsymbol{H}$,

$$\text{SCI}_{er} = \boldsymbol{U}^T_{r\times n}\boldsymbol{D}_{n\times n}\boldsymbol{U}_{n\times r} \quad \text{SCI}_{ra} = \boldsymbol{U}^T_{r\times n}\boldsymbol{G}_{n\times n}\boldsymbol{U}_{n\times r}$$

El cociente es ahora una matriz

$$\frac{\text{SCI}_{\boldsymbol{er}}}{\text{SCI}_{\boldsymbol{ra}}} = (\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{G}\boldsymbol{U})^{-\boldsymbol{1}}(\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{D}\boldsymbol{U}) \tag{3.21}$$

La valoración de dicha matriz puede realizarse en función de su traza. Una función criterio que se puede utilizar análogamente a la ecuación (3.21) para $p > 2$. Se tendrá:

$$J(\boldsymbol{U}) = Traza\{(\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{G}\boldsymbol{U})^{-\boldsymbol{1}}(\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{D}\boldsymbol{U})\} \tag{3.22}$$

Buscar la matriz $\boldsymbol{U}$ que maximice la traza de $(\boldsymbol{U^T GU})^{-1}(\boldsymbol{U^T DU})$ equivalente a resolver [71]:

$$\frac{\partial}{\partial U}(\boldsymbol{U^T GU})^{-1}(\boldsymbol{U^T DU}) = \boldsymbol{0} \implies \boldsymbol{DU(U^T GU) - (U^T DU)GU = 0}$$

al multiplicase por $(\boldsymbol{U^T GU})^{-1}$, y teniendo como supuesto que $\boldsymbol{G}$ es invertible, se tiene que:

$$\boldsymbol{DU - \lambda GU} = 0 \quad \Leftrightarrow \quad (\boldsymbol{D - \lambda G})\boldsymbol{U} = 0 \quad \Leftrightarrow \quad (\boldsymbol{G^{-1}D - \lambda I})\boldsymbol{U} = 0$$

de lo que se deduce que los pesos $\boldsymbol{U}$ son los autovectores de $\boldsymbol{G^{-1}D}$.

Reducción de dimensión y ADL. La cantidad de autovalores no nulos de $\boldsymbol{G^{-1}D}$ define la dimensión del espacio discriminante y por tanto es el número máximo de funciones discriminantes empleadas para representar la muestra.

Si $\boldsymbol{G}$ es no singular y $N - p \geq n$, donde p son los grupos y N las observaciones de dimensión n, se cumple que $r \leq \text{mín}\,(n, p-1)$ [72]. Para este caso, la restricción $N - k \geq n$, muestra que el número de patrones menos el de grupos debe al menos igualar a la dimensión de la muestra. Ahora bien, si $r \leq n$ el clasificador realiza una reducción de dimensión. Al número de componentes empleado para representar la muestra, se le conoce como *dimensión del espacio discriminante.*
La magnitud relativa (en porcentaje) de los respectivos autovalores λ_i, es un indicador de la importancia de cada eje discriminante, y puede calcularse como:

$$\frac{\lambda_i}{\sum_j \lambda_j} \,.100$$

3.4.2 Clasificadores de distribución libre o no paramétricos

Los clasificadores *no paramétricos* o *libres* son aquellos que utilizan técnicas de clasificación que no realizan hipótesis sobre las distribuciones de probabilidad subyacentes. Las más usadas son las siguientes:

- *Estimación del valor de la función de densidad.* Los clasificadores bayesianos usan el numerador de la ecuación:

$$p(C_j/\boldsymbol{x}) = \frac{p(\boldsymbol{x}/C_j)\cdot \pi_j}{p(\boldsymbol{x})}, \quad \text{donde} \;\; p(\boldsymbol{x}) = \sum_{i=1}^{k} p(\boldsymbol{x}/C_i)\cdot \pi_i \qquad (3.23)$$

siendo $p(C_j/\boldsymbol{x})$ el cálculo de las probabilidades de pertenencia de la muestra $\boldsymbol{x}$ a cada una de las clases C_j. Para construir un conjunto de k funciones discriminantes, lo que genera un método de clasificación que proporciona el mínimo error. En este caso si es imposible asumir una forma funcional para $p(\boldsymbol{x}/C_j)$, el clasificador bayesiano deberá utilizar una estimación de la función de densidad obtenida a partir de conjunto de patrones $\wp$ definido en la sección 3.4.2.

- **Estimación directa de la probabilidad a posteriori.** *El método de clasificación del vecino más próximo.* Establecida por [73, 74] a principio de los años 50, y formalmente enunciado por [75] en 1967 para ser usada como herramienta de clasificación de patrones, convirtiéndose en uno de los métodos de clasificación más usados [76]. Este método proporciona un enfoque basado en la estimación directa de la probabilidad a posteriori de cada clase, lo que genera reglas de clasificación sencillas, interpretables fácilmente y sólidamente justificadas.

- **Aprendizaje adaptativo.** Este tipo de técnicas están basadas en la selección de un conjunto representativo de prototipos que se utilizarán posteriormente para:
  Aproximación de funciones de densidad de probabilidad. Este tipo de aproximación emplea técnicas de cuantificación vectorial o LVQ (*Learning Vector Quantization*) [77].

  Aproximación de las fronteras de decisión. En este tipo de técnicas se utiliza el algoritmo DSM (*Decision Surface Mapping*) o Aproximación de Superficies de Decisión [78].

  En ambos casos, se trata de la construcción de un nuevo conjunto de referencia para la posterior clasificación considerando el vecino más cercano. La construcción del conjunto de referencia relacionado con el aprendizaje es homologa a la que se implementa en una red neuronal tal y como se verá en el capítulo 5.

- **Arboles de decisión.** Mediante una organización de tipo árbol, se aproxima a una solución organizada jerárquicamente del espacio de

representación usando datos categóricos o continuos. Los algoritmos más conocidos son el *CART* [79] y *C4.5* [80].

3.5 Selección del subconjunto de características

La selección de características se utiliza para obtener un subconjunto representativo de características. Los criterios de selección de características supervisadas se dividen en tres tipos principalmente: de filtro (*filter*), de envoltorio (*wrappers*) e incrustados (*embedded*).

Los primeros se utilizan para encontrar el mejor subconjunto características del conjunto original de características. Los métodos de filtrado parecen ser menos óptimos, sin embargo, son buenos en la selección de un gran subconjunto de datos, no dependen del algoritmo de clasificación, y su costo computacional es menor para grandes conjuntos de datos [81].

Los criterios de envoltorio (*wrapper*) utilizan la predicción del rendimiento del algoritmo de aprendizaje para la selección de las características. Mejoran los resultados de los predictores correspondientes, y logran mejores tasas de reconocimiento, incluso superando a la de los filtros, pero dependen del algoritmo de clasificación, y para un conjunto de datos grandes, el costo computacional es mayor [81].

Por último, los criterios incrustados (*embedded)* se denominan así porque forman parte del propio algoritmo de clasificación, es decir que las características se seleccionan durante la construcción del clasificador. Estos se basan en la evaluación del desempeño de la métrica calculada directamente de los datos, sin referencia directa a los resultados de los sistemas de análisis de datos. En ellos hay una unión de las técnicas de selección de características, con el proceso de aprendizaje para un algoritmo de aprendizaje determinado. Estos métodos son menos propensos al sobreajuste (*overfitting*), y también dependen del algoritmo de clasificación.

La Tabla 3.1 [82], describe las ventajas y desventajas y proporciona ejemplos, de cada una de las categorías de métodos de selección de características.

Tabla 3.1 Ventajas e inconvenientes de las categorías de métodos de selección de características

Método	Ventajas	Inconvenientes	Ejemplos
Filters	• Independiente del clasificador. • Más bajo costo computacional que los de tipo envoltorio. • Rápido. • Buena capacidad de generalización.	No interactúa con el clasificador.	Consistency-based CFS Interact ReliefF Md Information Gain mRMR
Embedded	• Interactúa con el clasificador. • Más bajo costo computacional que el de envoltorio. • Captura dependencias de las características.	La selección depende del clasificador.	FS-Percepton SVM-RFE
Wrapper	• Interactúa con el clasificador. • Captura dependencias de las características.	Es costoso a nivel computacional. Riesgo de sobreajuste. La selección depende del clasificador.	Wrapper-C4.5 Wrapper SVM

El criterio óptimo para evaluar un conjunto de datos de cara a un problema de clasificación sería el error Bayesiano cometido:

$$E(S) = \int_{\vec{S}} p(\vec{S}) \left(1 - \max_i \left(p(c_i|\vec{s})\right)\right) d\vec{S}, \tag{3.24}$$

donde $\vec{S}$ es el vector de características seleccionadas, y $c_i \in C$ es una clase de entre todas las clases de C presentes en el conjunto de datos.
Debido a que la función $máx(\cdot)$ no es lineal, el criterio de minimizar el error Bayesiano no es una función viable. Se pueden encontrar en la literatura varias cotas de error Bayesiano. Una de ellas obtenida en [83], es:

$$E(\vec{S}) \leq \frac{H(C|\overrightarrow{S})}{2} \tag{3.25}$$

Esta cota (3.25) está relacionada con la información mutua porque ésta última puede expresarse como:

$$E(\vec{S}; C) = H(C) - H(C|\overrightarrow{S}) \tag{3.26}$$

donde $H(\vec{C})$ es la entropía de las etiquetas de las clases, que no depende del espacio de características $\vec{S}$. Por tanto, la maximización de la información mutua es equivalente a la maximización de la cota superior del error Bayesiano. De igual forma, también existe una cota inferior, esta última

obtenida en [84], que también está relacionada con la información mutua.

Con frecuencia se piensa que un aumento de la dimensión del espacio de características será siempre beneficioso para discriminar entre dos clases. Curiosamente ocurre exactamente lo contrario, dando lugar a un problema conocido como "*course of dimensionality*" o "maldición de la dimensionalidad", o el "*peaking phenomenon*" llamado también el "*fenómeno del máximo*" [85]. Este problema se presenta como una reducción de la eficacia de un clasificador al añadir nuevas características a los vectores de entrenamiento cuando su cantidad es pequeña en relación al número de características. Se debe tener en cuenta en este problema que para definir un clasificador en un espacio de características de alta dimensionalidad es necesario estimar un número de parámetros comparable a la dimensión del espacio. Claro ejemplo es el de un clasificador lineal, donde será necesario estimar $m + 1$ parametros en un espacio de características *m*-dimensional.

Por consiguiente, aunque el clasificador realice el proceso de separación de datos de entrenamiento satisfactoriamente, la fiabilidad en la estimación de los parámetros será baja, debido a que se estimarán muchos parámetros con un número muy limitado de vectores de entrenamiento. La anterior limitación hace que el clasificador que se construya con esta metodología tenga, por consiguiente, una baja capacidad de generalización.

El problema de la maldición de la dimensionalidad "*course of dimensionality*" motiva el uso de técnicas de reducción de la dimensionalidad del espacio de características, cuando el número de características usadas para el diseño del clasificador es mucho mayor al número de vectores de entrenamiento. Existen otros motivos para reducir la dimensión del espacio de características hasta un mínimo razonable como son la reducción del coste computacional de los algoritmos de entrenamiento y test, la eliminación de la correlación entre características, o la selección de las características más relevantes para la clasificación.
Así, aunque el problema de la maldición de la dimensionalidad no exista, se puede suponer que el uso de un subconjunto de características con mejor capacidad de discriminación entre clases, mejorarán tanto el coste computacional del algoritmo de clasificación como el rendimiento del mismo.

3.5.1 Métodos de Filtrado

En este enfoque [86], se descartan los atributos irrelevantes para la clasificación antes de que esta ocurra. Así, a través de este paso de

preprocesamiento de datos se usan los aspectos generales del conjunto de entrenamiento para seleccionar o extraer unas características, y así mimo excluir otras. Por tanto, los métodos de filtro no dependen de los algoritmos de clasificación, y consiguientemente, se podrán combinar con cualquiera de dichos algoritmos, sin que se use el subconjunto de características obtenido mediante el filtrado para clasificación.

La técnica estadística de PCA (Análisis de Componentes Principales), ayuda a generar combinaciones lineales de los elementos originales, cuya matriz de transformación está formada por vectores que son ortogonales en el espacio original. A pesar de lo anterior, es necesario cierto criterio para seleccionar un subconjunto de los datos transformados, para reducir la dimensionalidad del espacio de características original.

De una manera empírica, mediante PCA se han logrado reducir la dimensionalidad en un amplio espectro de problemas de aprendizaje. En [87] se describen las garantías teóricas de los métodos de esta forma, cuando la función objetivo es una intersección de "semi-espacios" y las muestras son elegidas a partir de una distribución relativamente buena. El método de Análisis de Componentes Independientes (***ICA***) [88], relacionado con el anterior, incorpora ideas similares, pero insistiendo en la independencia de las nuevas características en lugar de su ortogonalidad. En [66] se muestra como PCA apoya el proceso de reducción de características para la clasificación de conexiones en redes de datos usando técnicas de proyección no lineales.

3.5.1.1 Análisis de Componentes Principales - PCA

PCA es técnica de extracción de características que permite identificar patrones en los datos, y expresarlos de tal manera que se puedan destacar sus semejanzas y diferencias. Esta metodología de análisis es muy utilizada para extraer información relevante de un conjunto inicial de variables correlacionadas transformándolas en variables no correlacionadas, con el objeto de identificar patrones y estructuras, y cuantificar la importancia de cada una de ellas a la hora de determinar la variabilidad en dicho conjunto de datos. Dado que los patrones pueden ser difíciles de encontrar en datos de alta dimensionalidad, donde no es posible la representación gráfica, PCA es una poderosa herramienta para hacer dicho análisis de datos. Otro propósito de PCA es que una vez que se han encontrado los patrones, se pueden comprimir los datos, mediante la reducción del número de dimensiones, sin mucha pérdida de información.

Funcionamiento de PCA. Se emplea para estudiar la relación existente entre un grupo de variables correlacionadas entre sí que contienen una información común. Como ya se ha dicho anteriormente, el objetivo principal es reducir la dimensión del conjunto de datos de entrada intentado mantener la mayor cantidad de información posible, para poder analizarlos de forma más fácil y que, en etapas posteriores, como clasificadores o regresores, se puedan simplificar los criterios de decisión.

Este método, realiza en primer lugar una transformación lineal de los datos en un nuevo sistema de coordenadas ortogonales. Los vectores de proyección de los datos en el nuevo espacio son las direcciones de máxima varianza de los datos de entrada. Mientras que las nuevas variables resultantes de proyectar los datos de entrada sobre los vectores de proyección se llamarán componentes principales ("*Principal Component*", PC). Estas nuevas variables son combinaciones lineales de las variables iniciales y se van construyendo según el orden de importancia en cuanto a la variabilidad total que explican de los datos iniciales.

La técnica de PCA se define matemáticamente como una transformación lineal ortogonal que transforma un conjunto de variables iniciales en un nuevo sistema de coordenadas de manera que la mayor varianza se sitúa en la primera coordenada (llamado el primer componente principal), la segunda mayor varianza en la segunda coordenada, y así sucesivamente. Aquí merece la pena destacar el concepto de que una mayor información se relaciona con el concepto de una mayor variabilidad. A mayor varianza se considera que existe mayor información.

La base matemática que se utiliza para desarrollar el PCA es el álgebra lineal. PCA está íntimamente relacionado con la técnica algebraica de la descomposición en valores singulares ("*Singular Value Decomposition*", *SVD*) [89]. El PCA tiene dos propiedades muy importantes que hacen de este método de reducción de la dimensionalidad tan popular [30]: (1) las componentes principales obtienen secuencialmente la máxima variabilidad o varianza de X, por lo que se garantiza la mínima perdida de información (en el sentido de error de reconstrucción), y (2) las componentes principales obtenidas son ortogonales entre sí, facilitando su posterior procesado, ya que pueden tratarse independientemente.

PCA trata de solucionar dos problemas comunes en el análisis de datos: (1) la dificultad para identificar la relación entre las variables cuando se emplea un número elevado de ellas, lo cual hace necesario reducir el número de variables (lo que se denomina "reducción de dimensión"), y (2) que muchas veces dichas variables pueden estar correlacionadas entre sí, siendo difícil

visualizar e identificar las relaciones entre ellas.

Los datos de entrada estarán dispuestos en una matriz $X \in \Re^{mxn}$, donde m es el número de variables de entrada y n es el número de observaciones. Mientras que los datos proyectados estarán en una matriz $Y \in \Re^{mxn}$, con p variables salida y n observaciones, donde el número de variables de salida es menor que el de entrada $(p < m)$. Cada vector $\mathbf{x}_i = [x_1, x_2, \dots, x_m]^{T'}$ donde T' indica la transposición del vector o la matriz, contendrá todas las variables de entrada asociadas a una observación i, e $\mathbf{y}_i = [y_1, y_2, \dots, y_m]^T$ todas las variables de salida.

El objetivo de PCA será entonces encontrar una matriz $H \in \Re^{mxp}$ que transforme linealmente el espacio de los datos de la entrada (en X), en otra matriz Y con un número menor de variables, aplicando una combinación lineal de las variables originales de modo que se proyecten sobre las direcciones de máxima varianza de los datos, y se conserve así la máxima cantidad de información. Estas direcciones de máxima varianza estarán definidas por los p vectores de proyección, $\mathbf{u}_k$,que se encontrarán en las columnas de H.

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_m \end{bmatrix} \underset{Y = H^T X}{\overset{PCA}{\rightarrow}} Y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_p \end{bmatrix}, \quad (m > p)$$

Durante este trabajo se llamarán "componentes principales" a las variables de salida $\mathbf{y}_k = \mathbf{u}_k^T \boldsymbol{X}$, mientras que los vectores de proyección $\mathbf{u}_k$ serán los coeficientes o cargas de la *k-esima* componente principal.

En principio, H puede tener hasta $p = m$ vectores de proyección, uno por cada variable. Sin embargo, habitualmente se reducirá esta dimensión conservando únicamente las proyecciones con mayor varianza, lo que produce una pérdida de información. Con el objetivo de minimizar esta pérdida el algoritmo del PCA consigue concentrar la mayor parte de la varianza en un número reducido de variables del nuevo espacio. De esta forma se pueden eliminar aquellas variables del espacio de salida que contengan menos varianza y así perder la menor cantidad posible de información. Como premisa, se busca que la mayor cantidad de información quede contenida en el menor número posible de variables.

Seguidamente, se obtiene la matriz de varianzas y covarianzas, Ω. Se emplea

la matriz de covarianzas, Ω, cuando las variables iniciales están expresadas en desviaciones frente a su media y es la matriz de correlaciones, ρ, cuando las variables están tipificadas. Así Ω es matriz cuadrada de dimensión pxp que posee en su diagonal principal las varianzas σ_i^2 de cada variable inicial x_i, y en los elementos no diagonales las correspondientes covarianzas σ_{ij}.

$$\Omega = \begin{bmatrix} \sigma_i^2 & \dots & \sigma_{1p} \\ \vdots & \ddots & \vdots \\ \sigma_{p1} & \dots & \sigma_p^2 \end{bmatrix}$$

Las p variables finales o "componentes principales" $(y_1, y_2, \dots, y_p)$ son el objetivo de PCA, siendo estas variables un reflejo de la variabilidad del conjunto inicial sin estar correlacionadas. Así,

$$y_j = a_{j1}x_1 + a_{j2}x_2 + \cdots + a_{jp}x_p \qquad (j = 1,2, \dots, p)$$

para cada una de las p variables y_j y par cada observación muestral t.

Teniendo en cuenta lo anterior, el primer componente principal (y_j) puede expresarse como:

$$\begin{bmatrix} y_{11} \\ \dots \\ y_{1n} \end{bmatrix} = \begin{bmatrix} x_{11} & \dots & x_{p1} \\ \vdots & \ddots & \vdots \\ x_{1n} & \dots & x_{p1} \end{bmatrix} \begin{bmatrix} a_{11} \\ \dots \\ a_{1p} \end{bmatrix}$$

$$y_1 = \mathbf{X}\,\boldsymbol{a}_1$$

Donde, y_1 es el vector de dimensiones $(nx1)$ y el primer componente principal, $\mathbf{X}$ será la matriz de dimensión $(nx1)$ de variables iniciales en columnas, y $\boldsymbol{a}_1$ el vector de constantes $(px1)$ para cada y_j. Tendremos entonces que el vector $\boldsymbol{a}_1$ es el encargado de relacionar las variables originales con cada una de las variables transformadas o componentes principales.

Consecuentemente se podrán expresar matricialmente las variables transformadas, en función de los autovectores y de las variables iniciales, como:

$$Y' = \boldsymbol{A}'\,\mathbf{X}' \qquad (3.27)$$

donde Y' denota la matriz traspuesta de componentes principales con

dimensión (pxn) , $\boldsymbol{A'}$ será la matriz traspuesta de autovectores con dimensión (pxp) y $\mathbf{X'}$ la matriz traspuesta de variables originales transformadas, bien sea en desviaciones respecto a la media o tipificadas, con dimensión (pxn). Así:

$$\begin{bmatrix} y'_1 \\ \dots \\ y'_p \end{bmatrix} = \begin{bmatrix} a'_1 \\ \dots \\ a'_p \end{bmatrix} \begin{bmatrix} x'_1 \\ \dots \\ x'_p \end{bmatrix}$$

$$\begin{bmatrix} t_{11} & \dots & t_{1n} \\ \vdots & \ddots & \vdots \\ t_{p1} & \dots & t_{pn} \end{bmatrix} = \begin{bmatrix} a_{11} & \dots & a_{1p} \\ \vdots & \ddots & \vdots \\ a_{p1} & \dots & a_{pp} \end{bmatrix} \begin{bmatrix} x_{11} & \dots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{p1} & \dots & x_{pn} \end{bmatrix}$$

Por lo tanto, el elemento $n - esimo$ del primer componente principal Y_1 se definirá como:

$$y_{1n} = a_{11}x_{1n} + a_{12}x_{2n} + \cdots + a_{1p}x_{pn}$$

Esta relación también puede ser expresarse a la inversa, de tal forma que las variables iniciales estén representadas en función de los componentes iniciales. Para esto, se parte de la relación (3.27):

$$Y' = \boldsymbol{A'}\,\mathbf{X'}$$

y se multiplica a ambos lados por $[\boldsymbol{A'}]^{-1}$, así:

$$[\boldsymbol{A'}]^{-1}Y' = [\boldsymbol{A'}]^{-1}\boldsymbol{A'}\,\mathbf{X'} \Longrightarrow\ \mathbf{X'} = [\boldsymbol{A'}]^{-1}Y'$$

teniendo en cuenta que $\boldsymbol{A}$ es ortogonal (como es la matriz $\boldsymbol{A}$ de autovectores), es decir es una matriz cuadrada cuya matriz inversa coincide con su matriz traspuesta, entonces $\boldsymbol{A'^{-1}}$=$\boldsymbol{A'}$, con lo que se cumple que $\boldsymbol{X'} = \boldsymbol{A'}\,\boldsymbol{Y'}$.

Por otro lado, si se utilizan los datos originales de las desviaciones respecto a la media, se debe añadir la media a los datos originales, es decir: $\boldsymbol{X'^{*}} = (\boldsymbol{A'}\,\boldsymbol{Y'}) + \boldsymbol{\mu}$, donde $\boldsymbol{X'^{*}}$ representa la matriz traspuesta de variables originales, sin transformar con dimensión (pxn), y $\boldsymbol{\mu}$ el vector columna de p medidas muestrales con dimensión (pxn).

Se debe tener presente que el autovalor λ_j corresponde a la varianza del

componente principal y_j , que es definido por medio del autovalor a_j :

$$Var(y_j) = \lambda_j$$

Como la matriz Λ de autovalores es diagonal, la variabilidad de los componentes principales estará dada por:

$$\sum_{i=1}^{p} Var(y_i) = \sum_{i=1}^{p} \lambda_i = traza\ (\Lambda) = \sum_{i=1}^{p} Var(x_i)$$

La suma de las varianzas de las variables originales x_i coincide con la suma de las varianzas de los componentes principales (y_i), y la suma de los autovalores de la matriz de covarianzas muestral Ω. Si las variables están tipificadas en vez de estar expresadas mediante desviaciones respecto de la media, la matriz de covarianzas será igual a la matriz de correlaciones, por lo que con lo que se tiene que $traza\ (\Omega)\ traza\ (P) = p$, y la proporción de la componente principal i-ésima en la variabilidad total será de λ_i/p.

Así se tienen las bases para calcular el porcentaje de varianza total por la componente principal i-ésima:

$$\frac{\lambda_i}{\sum_{i=1}^{p} \lambda_i} = \frac{\lambda_i}{\sum_{i=1}^{p} Var(x_i)}$$

pudiendo determinar el porcentaje de la variabilidad total recogido por las m primeras componentes principales $(m < p)$:

$$\frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{p} \lambda_i}$$

3.5.2 Métodos de Envoltura (*wrapper*)

El segundo enfoque para la selección de características son los métodos de Envoltura [90, 86]. Estos utilizan la clasificación como subrutina, en lugar de hacerse como un postprocesamiento (en el caso de los filtros). Su funcionamiento se centra en la búsqueda de subconjuntos en el espacio de características que produzcan resultados óptimos en la clasificación, ejecutando internamente la función clasificadora en cada alternativa. Después de realizado este proceso, la selección de características se hace

utilizando como criterio la estimación de la precisión del clasificador, seleccionándose aquellas características que arrojaron los mejores resultados.

Existen autores como [86] y [91] que defienden el uso de los métodos de envoltura, mostrando la implementación de un método de envoltura para mejorar el comportamiento de inducción de los árboles de decisión. Además, en [86] se muestran estudios donde se compara el uso y los resultados de los métodos de filtro contra los métodos de envoltura.

En [90] se muestra como los métodos de envoltura logran proporcionar soluciones más precisas que los métodos de filtrado al problema específico de selección de características. Sin embargo, el principal inconveniente de los métodos de envoltorio frente a los de filtro es su coste computacional, resultado de llamar al algoritmo de clasificación para evaluar cada conjunto de características consideradas.

Un método innovador es propuesto por De la Hoz [92], donde se presenta un proceso de selección de características basado en optimización multiobjetivo, mediante una técnica de envoltura que utiliza el algoritmo NSGA-II. La técnica entrena y clasifica una estructura GHSOM (Mapa Auto-organizativo de Jerarquía Creciente), tomando los datos a partir del conjunto NSL-KDD. La selección de características se realiza mediante la implementación del algoritmo NSGA-II [92], que toma la estructura de un cromosoma, constituido por genes binarios, que representan la activación o no de cada característica. Tras cada iteración se calculan los índices de *Jaccard* [92] (como criterio de disimilitud de los grupos de datos) que permiten identificar cuáles son las mejores soluciones (conjuntos de características) a partir de los frentes de **Pareto** obtenidos, los cuadrados azules de la Figura 3.4 describen globalmente esta técnica *wrapper*.

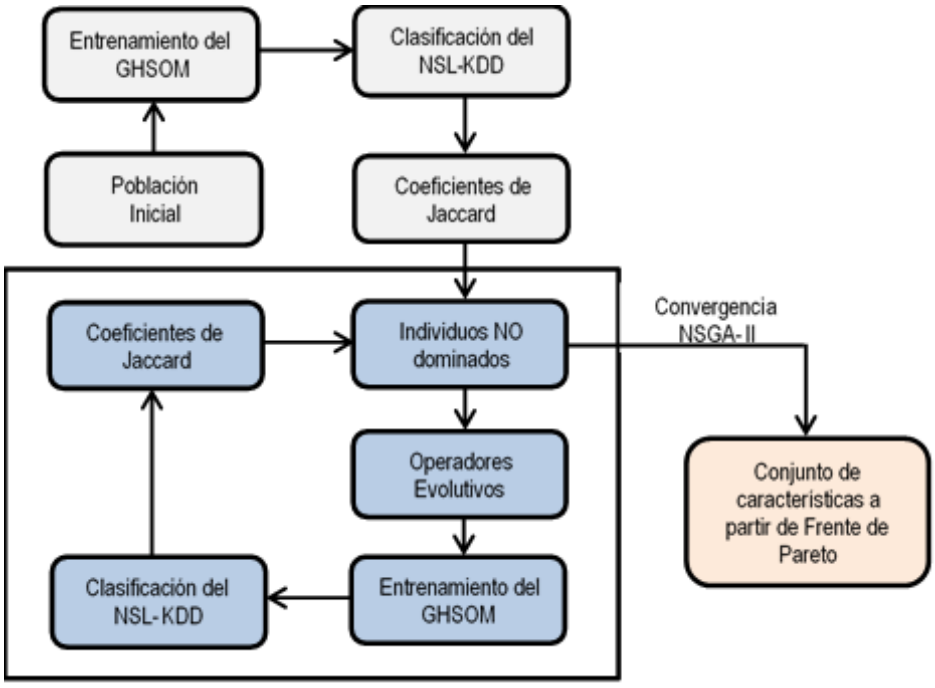


Figura 3.4. Esquema general de la técnica de selección de características de [92].

3.6 Parámetros de valoración del rendimiento de un clasificador

El principal objetivo de un clasificador está en la correcta asignación de una etiqueta a un patrón de entrada. Cuando se utiliza clasificación binaria, donde los datos pertenece a alguna de las clases posibles (por ejemplo, patrones positivos o negativos), el clasificador puede cometer alguno de estos errores: clasificar como positivo un patrón que verdaderamente era negativo o viceversa. Estas posibles situaciones se detallan en [71] y se muestran en la Tabla 3.2. En ella se muestran los resultados de la comparación del conjunto de test con la etiqueta original, dando lugar a los VP o los (Verdaderos Positivos) que es cuando los dos coinciden en valor positivo, los FP (Falso Positivo) ocurrencia que se da cuando el test da positivo pero la etiqueta original era negativa, los FN o (Falso Negativo) cuando el test da negativo mientras que la etiqueta original era positiva, por último se muestran os VN (Verdadero Negativo) refiriéndose al caso cuando los dos coinciden en valor negativo.

Tabla 3.2. Posibles resultados del test en función de la etiqueta

		Etiqueta		
		Positiva	Negativa	
Test	Positivo	VP	FP	Valor predictivo positivo
	Negativo	FN	VN	Valor predictivo negativo

Sensibilidad Especificidad

A nivel estadístico se conoce como función de clasificación a las medidas del desempeño de una prueba de clasificación binaria, donde dicha prueba se aplica usando la sensibilidad (*sensitivity*) y especificidad (*specificity*). La primera de ellas también llamada tasa de recuperación (*recall rate*) se encarga de medir la proporción de "verdaderos positivos" que son correctamente identificados como tales. La especificidad mide la proporción de "verdaderos negativos" que se han identificado correctamente. A partir de las definiciones anteriores, se tiene que:

$$\text{Sensibilidad} = \frac{\text{VP}}{\text{VP} + \text{FN}} \tag{3.28}$$

$$\text{Especificidad} = \frac{\text{VN}}{\text{VN} + \text{FP}} \tag{3.29}$$

Dos métricas complementarias son la Tasa de Falsos Positivos (TFP), y la Tasa de Falsos Negativos (TFN).

$$\text{TFP} = \frac{\text{FP}}{\text{VN} + \text{FP}} = 1 - \text{Especificidad} \tag{3.30}$$

$$\text{TFN} = \frac{\text{FN}}{\text{VP} + \text{FN}} = 1 - \text{Sensibilidad} \tag{3.31}$$

Existen dos métricas muy utilizadas en la bibliografía, la primera de ellas es la precisión (*accuracy*) [93] definida como el grado de cercanía de las mediciones de una cantidad al valor de la magnitud real. La precisión o valor predictivo positivo se define como la proporción de VP contra todos los resultados positivos, definida en:

$$\text{Precisión} = \frac{\text{VP}}{\text{VP} + \text{FP}} \tag{3.32}$$

La segunda, es la exactitud que define la proporción de resultados verdaderos (tanto verdaderos positivos como verdaderos negativos) en la población. Una exactitud del 100% significa que los valores medidos son exactamente los mismos que los valores dados. La exactitud se define en la ecuación (3.33).

$$\text{Exactitud} = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{FP} + \text{FN} + \text{VN}} \tag{3.33}$$

Se puede presentar el caso que un clasificador tenga valores cercanos al 100% de sensibilidad y cercanos al 0% de especificidad, en este ejemplo puntual este clasificador no podrá discernir entre las clases, tomando cualquier patrón como positivo. Lo anterior conduce a que este clasificador se compare con un clasificador aleatorio, ya que su precisión estará alrededor del 50% arrojando un muestreo sin preponderancia suficiente de ninguna de las dos clases. Un clasificador ideal será aquel donde los valores de Sensibilidad, Especificidad y Precisión de manera simultánea tengan valores altos y no aquel que cuente con valores altos de manera separada en alguno de estos indicadores.

3.7 Conclusiones

En este capítulo se hizo un estudio detallado del aprendizaje estadístico, pasando por el aprendizaje supervisado y No supervisado. Se detallaron los métodos Kernel, tanto las SVM para clasificación como para regresión. Así mismo, se aborda el tema de los clasificadores estadísticos, mostrando la fundamentación básica de lo que es un clasificador. Se mostró la definición de muestra de aprendizaje y de validación que próximamente se utilizará en la etapa de configuración experimental de esta tesis, de igual forma se mostraron los clasificadores paramétricos y de distribución. Con miras a darle solidez a esta tesis, en la Sección 3.5 se hizo un análisis de los métodos de filtrado y de envoltura, haciendo énfasis en el análisis de componentes principales – PCA, método fundamental para la implementación de esta tesis.
Si se asume que todos los errores de clasificación son igualmente costosos y que no hay preferencia ni penalización para un determinado conjunto de patrones, un buen clasificador debería obtener un alto nivel de precisión global, así como un aceptable nivel de precisión para cada clase. Con el objetivo de hacer un análisis del rendimiento del clasificador planteado, en este capítulo se mostró con detalle los parámetros de valoración que se tienen en cuenta al evaluar los clasificadores estadísticos.

Las métricas usadas nos permiten hacer una clasificación de patrones. Estas, obtienen un alto porcentaje de acierto en la clasificación implementada, algo de suma importancia en el área de la detección de anomalías en redes de computadores.

Capítulo 4. Redes Neuronales Artificiales

En el presente capítulo se expondrán principales aspectos de las redes neuronales, haciendo hincapié en las arquitecturas utilizadas en el desarrollo de los métodos de clasificación que se proponen en este trabajo. A continuación, se consideran los contenidos relacionados con el aprendizaje en las redes neuronales. Para concluir se explican más detalladamente las redes neuronales autoorganizables, y las redes SOM (*Self-Organizing Maps*) describiendo su funcionamiento y el algoritmo que permite implementarlas.

4.1 Generalidades

El ser humano lleva mucho tiempo preguntándose acerca del origen de la mente o del funcionamiento del cerebro. Desde que se sentaron las bases de la lógica hasta hoy el tema de las funciones mentales y su relación con el funcionamiento fisiológico del cerebro es un campo de investigación fascinante. Estos temas han dado para hacer múltiples estudios tanto en biología como en otras áreas donde el cerebro se toma como base para desarrollar aplicaciones reales de su forma de procesamiento. Así, las *redes neuronales artificiales* surgen como modelos computacionales que tienen al cerebro humano como fuente de inspiración. En esencia, el concepto parte de un conjunto de procesadores denominados neuronas artificiales interconectados de acuerdo a una arquitectura previamente definida. De manera tal que la información fluye en paralelo y distribuyéndose por varias neuronas. Estas características permiten que las redes neuronales artificiales tengan un gran poder computacional, ya que son capaces de procesar grandes cantidades de datos y analizarlos simultáneamente desde distintas perspectivas.

El valor de las redes neuronales empieza a tomar fuerza en los años 80 en el que se realizan las primeras publicaciones que ratifican su uso y dan veracidad de sus resultados. Una de las publicaciones más influyentes fue "*Neural networks and physical systems with emergent properties*" [94] de Hopfield, donde se dieron las primeras razones formales para justificar el funcionamiento de las redes neuronales y por lo tanto dio legitimidad a su uso. Otro de los resultados que dieron validez a las redes neuronales es que problemas combinatorios de optimización, como el problema del viajante, fueron formulados y resueltos en términos de una función de energía de una red, que se minimiza cuando la red alcanza un estado estable.

Probablemente las características más importantes en la dinámica de una red neuronal son las llamadas propiedades emergentes. En palabras de Hopfield [94] "*las propiedades emergentes nacen de la interacción de un gran número de elementos, entonces son la consecuencia de relaciones microscópicas que parecen tener vida propia*".

Actualmente las redes neuronales artificiales se pueden simular en software o ser implementadas en hardware. La implementación en hardware en su totalidad y de manera eficaz y eficiente es hoy día el gran reto, ya que es la única manera de aprovechar al máximo todas las capacidades de las RNA (Redes Neuronales Artificiales). Al mismo tiempo, miles de investigadores en diversos campos, como la neurociencia, la psicología, la medicina, las matemáticas, la física y la ingeniería han abordado un número importante de problemas del mundo real utilizando redes neuronales.

4.2 Arquitecturas de las redes neuronales

Teniendo en cuenta que las redes neuronales artificiales son un ensamble de neuronas "artificiales", estas pueden representarse por medio de un grafo dirigido. Es decir, mediante un conjunto de vértices unidos por flechas donde la cantidad de vértices es un conjunto de neuronas. A las conectividades entre las neuronas se le denomina arquitectura de la red.

Teniendo en cuenta lo anterior las arquitecturas de las RNA se pueden dividir en dos grandes categorías: (1) las redes "hacia delante" (*feedforward*) en las cuales no existen ciclos; y (2) las redes de retroalimentación (*feedback*) en las cuales sí existen ciclos y por lo tanto la red se retroalimenta. Las redes hacia-adelante son redes con varias capas donde solo existe conexión entre capas consecutivas y no tienen memoria ya que la respuesta de una de estas redes a un dato de entrada dado es independiente de los estados previos de la red. Las redes con retroalimentación son sistemas dinámicos

donde por cada dato de entrada, las respuestas de las neuronas son computadas por medio de las conexiones de retroalimentación ocasionando una modificación en los vectores de pesos de las neuronas. De esta manera, la red se encuentra en un nuevo estado, repitiendo el proceso hasta alcanzar algún tipo de equilibrio o convergencia.

4.3 El aprendizaje de una red neuronal artificial

Toda red neuronal debe tener la capacidad de aprender independientemente de su arquitectura. El concepto de aprendizaje se relaciona con actualización de los pesos sinápticos de las neuronas con el fin específico de mejorar la forma en que realiza cierta tarea concreta. Para poder hacer efectivo este aprendizaje, se tiene en cuenta el llamado vector de pesos que representa la actualización iterativa de los pesos sinápticos a los que nos hemos referido. Con este proceso de aprendizaje, la red neuronal alcanza su desempeño óptimo mediante la modificación de los vectores de pesos y es precisamente por esta razón que se considera a las redes neuronales sistemas adaptables.

La actualización de los pesos en las redes neuronales se puede hacer de varias formas, algunas de ellas utilizan información previamente dada mientras que otras deben aprender a establecer los pesos correctos a partir del conjunto de datos.

Aplicando lo anterior y lo expuesto con detalle en el capítulo 4, encontramos tres formas en las que una red neuronal puede aprender. La primera de ellas es *mediante el aprendizaje supervisado*, en el que se provee a la red las respuestas correctas para cada elemento del conjunto de entrenamiento. Un ejemplo típico de este tipo de aprendizaje es el "perceptron multicapa" [95]. Siendo el perceptron 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 o negativas de la clase se pueden separar mediante un hiperplano en el espacio de características. De manera análoga, el perceptron multicapa está compuesto por una capa de entrada, una de salida y una o más capas ocultas. La arquitectura del perceptron multicapa se caracteriza porque las conexiones entre neuronas son siempre hacia delante, es decir, van desde las neuronas de una determinada capa hacia las neuronas de la siguiente capa; no existen conexiones laterales ni conexiones hacia atrás. Lo anterior hace que la información se transmita desde la capa de entrada hacia la capa de salida.

En la segunda clase de redes neuronales se implementa el *aprendizaje no supervisado* [96], que no requiere de que se disponga de respuestas correctas

ya que es la red la encargada de explorar los datos y buscar correlación entre los posibles patrones que puedan existir. Para este caso encontramos los SOM, constituidos por una familia de redes neuronales que proyectan las muestras en una malla de unidades o neuronas que organizan sus vecindades de acuerdo con la estructura intrínseca de los datos.

Por último, tenemos las redes con *aprendizaje híbrido* [97] en el que se programa la neurona para que aprenda con una porción de los datos utilizando aprendizaje supervisado, mientras que los pesos restantes se determinan de manera no supervisada.

4.4 Redes Neuronales Autoorganizables

Las redes más utilizadas en tareas de agrupamiento de datos son aquellas que hacen uso del aprendizaje competitivo, siendo los mapas autoorganizables de Kohonen (SOM) [98], las redes *Adaptative Resonance Theory* (ART) [99] y las derivaciones de ambas, los modelos más difundidos.

En esta memoria, nos centramos en las redes SOM [98], y las Probabilistic Self-Organizing (PSOM) [100].

Aparte de estas redes, existen otras, la mejora de las SOM denominadas Growing Hierarchical Self-Organizing Maps (GHSOM) [101], y aquellas que implementan modelos constructivos como son Growing Neural Gas [102], Incremental Grid-Growing [103], Growing Self-Organizing Maps [104] y Density Based Growing Neural Gas [105].

4.5 Las redes SOM (Self-Organizing Maps)

En este apartado se presenta el modelo de red neuronal propuesta por T. Kohonen [98] denominado Self-Organizing Map (SOM), que constituye un algoritmo eficiente para llevar a cabo la visualización de grandes conjuntos de datos multidimensionales.

Se han presentado evidencias de que hay neuronas en el cerebro capaces de organizarse en muchas zonas de forma tal que la información captada del entorno a través de los órganos sensoriales se representa internamente en forma de mapas bidimensionales [106]. Aunque esta organización neuronal está predeterminada genéticamente, se presume que parte de ella se origina

mediante el aprendizaje, sugiriendo que el cerebro lograría tener la posibilidad de formar mapas topológicos de la información recibida del exterior. Es precisamente en estas ideas en las que se basa *T. Kohonen* para proponer un modelo de red neuronal con capacidad de formar mapas de características de manera similar a como ocurre en el cerebro donde un estímulo externo o datos de entrada por sí solo, con estructura propia y descripción funcional del comportamiento de la red, puede originar mapas topológicos de semejanza con los creados por el cerebro.

4.5.1 Funcionamiento de SOM

Una red SOM trata de establecer una correlación entre los datos de entrada y un espacio bidimensional de salida, creando mapas topológicos de dos dimensiones, con miras a que, si existen datos de entrada con características comunes, se activen neuronas situadas en zonas próximas de la capa de salida. Una representación visual de este procedimiento se muestra en la Figura 4.1. Corresponde a una arquitectura típica de un mapa auto organizado en las que neuronas de salida se disponen en forma bidimensional para representar los mapas de características. De ahí que los SOM también se les llame *Kohonen Feature Maps*.

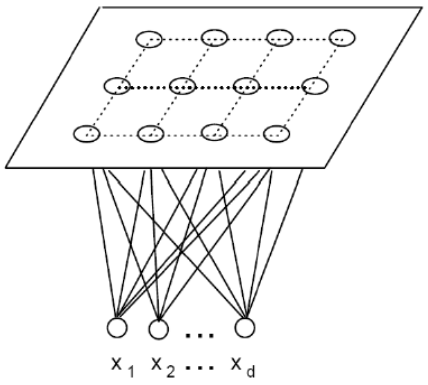


Figura 4.1. Arquitectura típica de un mapa auto organizativo [107]

Esta visualización del conjunto de datos es generada de forma automática y sin intervención humana, lo que categoriza al algoritmo SOM como un algoritmo de redes neuronales de aprendizaje no supervisado a los que nos hemos referido anteriormente, que no utiliza valores a priori, ni tiene retroalimentación. Este modelo es uno de los más populares que se utilizan en redes neuronales artificiales y pertenece a la categoría de redes con aprendizaje competitivo. En este aprendizaje, las células reciben de manera

idéntica la información de entrada sobre la cual compiten, siendo esta competencia una lucha por determinar cuál de las neuronas es la que mejor representa a un estímulo de entrada dado. Como resultado de esta competición, solo una neurona es activada en cada momento, como se muestra en la Figura 4.2.

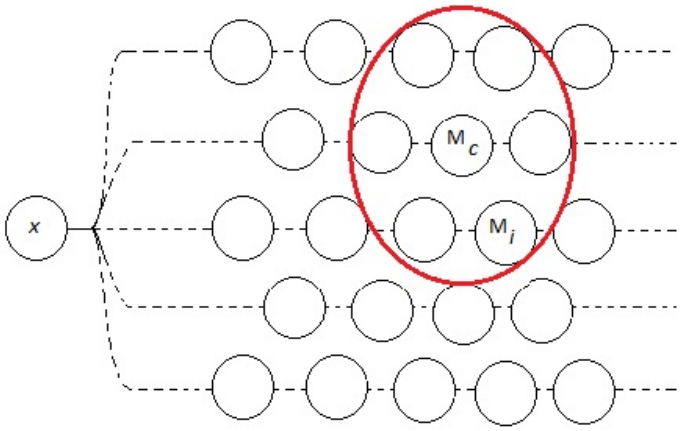


Figura 4.2. Neurona ganadora = Mc y Neurona vecina = Mi [107]

4.5.1.1 El algoritmo de SOM (Self-Organizing Maps)

"*La autoorganización es el proceso por medio del cual, en un sistema de unidades individuales, por medio de interacciones cooperativas, emergen nuevas propiedades en el sistema que trascienden a las propiedades de sus partes constitutivas*" [108].

Un mapa autoorganizativo (SOM), como red neuronal artificial de entrenamiento competitivo, cuenta con un mecanismo autoorganizativo en el que la neurona ganadora tiene potestad de modificar el vector de referencia de las unidades vecinas. La magnitud de la modificación está en función de la distancia física entre la neurona ganadora y cada una de las neuronas vecinas. En este sentido, el mecanismo de autoorganización en un SOM consiste en usar su capacidad de aprendizaje adaptable para representar la estructura topológica subyacente en el conjunto de datos de entrenamiento. Por tanto, la representación será posible debido a la autoorganización topográfica de las neuronas de acuerdo a las relaciones de similitud entre los datos representadas por la cercanía entre las neuronas y los vectores de referencia correspondientes. De lo anterior podemos concluir entonces que un SOM proporciona un mecanismo que brinda la posibilidad de producir automáticamente una representación del conjunto de datos en una estructura bidimensional en la que se haga evidente la emergencia de propiedades que ayuden a entender el orden geométrico subyacente en el conjunto de datos.

Una vez que el algoritmo de SOM ha convergido, el mapa de características, $\boldsymbol{\phi}$, obtenido en el espacio de salida, $\boldsymbol{M}$, muestra propiedades estadísticas relevantes del espacio de entrada, $\boldsymbol{X}$. Las siguientes características podrían destacar [109]:

- *Aproximación del espacio de entrada.* El mapa de características, $\boldsymbol{\phi}$, representado por el conjunto de vectores modelo $\{m_i\}$ en el espacio de salida, $\boldsymbol{M}$, da una buena aproximación del espacio de entrada $\boldsymbol{M}$.

- *Orden topológico.* Los mapas de características, $\boldsymbol{\phi}$, obtenidos a través del algoritmo SOM están clasificadas topológicamente en el sentido de que la localización espacial de un vector modelo de la red corresponda a un dominio particular o característica de los patrones de entrada.

- *Coincidencia de densidad.* Los mapas de características, $\boldsymbol{\phi}$, reflejan las variaciones en las distribuciones estadísticas de las entradas.

- *La selección de características.* Cuando se provee un conjunto de datos del espacio de entrada con una distribución no lineal, el mapa auto-organizativo es capaz de seleccionar un conjunto con las mejores características con el fin de aproximar la distribución subyacente.

Las principales aplicaciones de los mapas organizativos son: la visualización, el agrupamiento de datos, la interpolación o modelado de funciones, la clasificación (en algunos casos) y la cuantificación vectorial. Un SOM efectivamente crea un mapa del espacio de entrada que se puede utilizar, una vez entrenado, para asociar las nuevas entradas a las características de los vectores utilizados para el entrenamiento.

El algoritmo que describe el funcionamiento de un SOM se explica a continuación:

- Un SOM es un vector bidimensional de neuronas que se puede describir como sigue:

$$M = \{m_{1,1}, \ldots\ldots, m_{p,x}\} \tag{4.1}$$

- Una neurona es un vector que se puede representar como:

$$M_i = \{m_{i1}, \ldots\ldots, m_{in}\} \tag{4.2}$$

La neurona tiene las mismas dimensiones que los vectores de entrada, es decir *n*. La relación que existe entre una neurona y sus respectivas vecinas está determinada por la topología o estructura del mapa. En la Figura 4.3 se muestran dos topologías posibles usadas en las SOM, la hexagonal y la rectangular

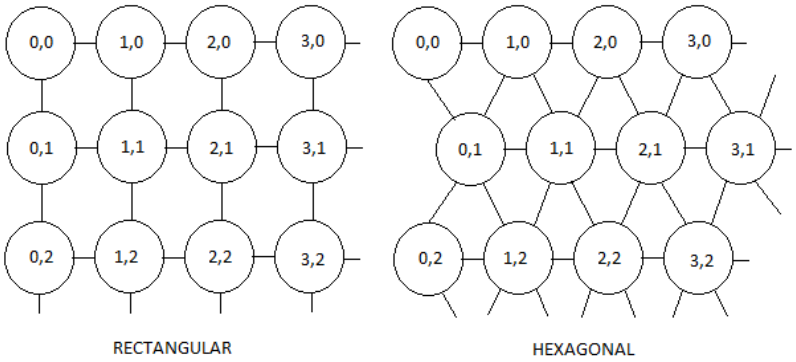


Figura 4.3: Posibles topologías de SOM [77]

En el algoritmo SOM básico, las relaciones topológicas entre los nodos (hexagonal o rectangular) y el número K de neuronas se fijan desde el inicio, definiendo la escala o la granularidad del modelo resultante y las distancias entre las unidades del mapa de acuerdo a la distancia Euclidiana entre los vectores de localización, sin embargo, en ocasiones es más práctico usar otras funciones de distancia.

En la ejecución del algoritmo de SOM existen varias etapas: el pre-procesamiento de datos, la inicialización, el entrenamiento, la visualización y, finalmente, la validación.

El ***pre-procesamiento*** de datos es la etapa de alimentación de datos extraídos de la red al SOM. Esto implica una etapa de eliminación de información no óptima para el procesamiento, por lo que se hace un análisis en primera instancia de un subconjunto de datos relevantes del modelo en cuestión.

En la gran mayoría de los casos los datos de entrada para hacer este pre-procesamiento deben normalizarse para tener una escala de 0 a 1 y poder usar la distancia como una medida de similitud.
El segundo punto a tener en cuenta en el algoritmo es la ***inicialización*** del SOM en sí. Entre las posibles maneras de inicializar el mapa podemos encontrar la inicialización aleatoria. Se consideró este enfoque aleatorio de inicialización por su sencillez, así como por la relativa sencillez para

inicializar los centros de los *clusters*. No es necesariamente el mejor enfoque, especialmente para la producción de una convergencia rápida a un estado estable [110, 111, 112, 113]. Esta inicialización puede realizarse de las siguientes formas:

- Componente aleatorio. Cada componente del vector de referencia se elige al azar del rango de valores observados en los datos.

- Selección aleatoria. Utiliza datos de la muestra, y tiene la ventaja de que las muestras se encuentran automáticamente en la misma parte del espacio de entrada de datos [114, 115].

- Perturbación aleatoria en torno a los valores medios. Inicializa los centros de los *clusters* perturbando ligeramente la media o el centro global de las entradas.

Con el fin de mejorar la solución obtenida por el SOM, los reinicios múltiples (inicialización aleatoria) se pueden utilizar en la primera y segunda etapa. Además de seleccionar el modelo que presenta un mejor rendimiento de clasificación [116, 117]. También se pueden crear alternativas diversas mediante el uso de diferentes métodos de inicialización en el objetivo de mejorar la solución obtenida por el SOM. Así, otra forma en la que se puede inicializar un SOM es ***basándose en los datos de análisis***. Esta forma de inicialización utiliza algunos métodos resultantes del análisis de datos estadísticos y de la clasificación datos para inicializar SOM. También pueden desarrollarse otros métodos para la inicialización del *cluster*. En la literatura se pueden encontrarse dos métodos principales para la inicialización del SOM según esta línea:

- Para la clasificación de datos: inicialización basada en k-medias (*k-means)*, que implica tres etapas. En la primera etapa, se utiliza el algoritmo *k-medias* para seleccionar $N \times M$ (es decir, el tamaño del mapa de características a formarse) centros de *cluster* a partir de un conjunto de datos. A continuación, se emplea una estrategia de asignación heurística para organizar los $N \times M$ puntos de datos seleccionados en una matriz $N \times M$ neuronal a fin de formar un mapa de características inicial. Si el mapa inicial no es lo suficientemente bueno, entonces se afina usando el tradicional algoritmo autoorganizativo de Kohonen [118] bajo un régimen de enfriamiento rápido en la tercera fase [119].

- Para el análisis de datos estadísticos: inicialización basada en

> proyección lineal de datos combinado con un método de proyección lineal con malla rectangular periódica. El método de proyección lineal más conocido es el Análisis de Componentes Principales (PCA), que ya se ha presentado en el capítulo anterior. Como se ha dicho, PCA es un enfoque no supervisado cuyo propósito es encontrar las características más sobresalientes de un conjunto de datos. Los *clusters* se inicializan por estar en el mismo espacio de entrada que es atravesado por los primeros vectores propios que corresponden a los mayores valores propios de los datos de entrada. El *algoritmo de re-constitución* se utiliza para estimar los valores de la muestra correspondiente a la celda vacía de la malla regular [120].

Acto seguido se debe hacer un ***entrenamiento***. Se trata de un proceso iterativo en el tiempo donde se van ingresando a la red muestras del conjunto de datos de entrada para que la misma las "aprenda". Este proceso suele ser lento y con un coste computacional alto ya que la razón del verdadero aprendizaje consiste en elegir una neurona ganadora utilizando el concepto de "medida de similitud" explicado, y con este ir actualizando los valores de los patrones de los vecinos. Como el proceso es iterativo, se repite hasta acotar el error y acercar a las neuronas a una representación optima de los datos de entrada.

Una representación matemática del proceso planteado de entrenamiento se detalla a continuación:

En cada momento t del proceso de entrenamiento, un vector de entrada $x\ (t)\ \Re^n$ es conectado a todas las neuronas en paralelo vía los vectores de referencia ω_i de cada neurona. Se produce entonces una competencia de las neuronas para saber cuál de ellas es capaz de representar de la mejor manera al dato de entrada $x(t)$.

La competencia consiste en que dado cualquier $x \in X$ hay que encontrar una neurona tal que su vector de referencia o vector prototipo ω_c cumpla con:

$$\|x - \omega_c\| = \min_{i=1,..,N} N\{\|x - \omega_i\|\} \qquad (4.3)$$

a la neurona ganadora η_c se le define como el nodo que mejor representa al dato x o *Best Matching Unit* (***BMU***). Cabe apuntar que el subíndice c es función de x para cada x existe un $\eta_c(x)$.

Si se presentará el punto en que este índice no esté bien definido, es decir

cuando para un dato x existan dos $\eta_e, \eta_d \in$ N tal que:

$$d(x,\eta_e) = mín\ \{d(x,\eta_i)/\ i\ = 1, \ldots., k\} =\ d(x,\eta_d),$$

la selección de un único $c(x)$ en este caso debe hacerse de manera aleatoria.

Tenemos entonces la siguiente notación:

$$x\ \approx\ \eta\ \Leftrightarrow\ \ \eta =\ \eta_{c(x)}$$

Para cada tiempo t se realiza la ecuación (4.3) de manera que se puede definir $c\ =\ c(t)$ tal que $x\ (t) \approx\ \eta_{c(t)}$, las neuronas que se encuentran dentro de una vecindad de $\eta_{c(t)}$ en el mapa bidimensional aprenderán de la misma entrada $x(t)$ como se muestra en la Figura 4.1.

Cada vez que se determina una neurona ganadora $\eta_{c(t)}$, la idea clave en el algoritmo de aprendizaje es que aquellas neuronas que se encuentran dentro de una vecindad $\eta_{c(t)}$ en el arreglo bidimensional también aprendan de la entrada $x(t)$. Para determinar la magnitud del aprendizaje de una neurona η_i en términos de la distancia con la neurona ganadora $\eta_{c(t)}$ se define la denominada *función de vecindad* (Figuras 4.5 y 4.6) que es de la forma:

$$h_{ci}(t) = h\ (\|r_{c(t)} -\ r_i\|, t)\ \in\ [0,1]$$

lo cual implica que el valor de la función depende de la distancia entre la neurona η_i y la neurona ganadora $\eta_{c(t)}$ en el tiempo t. Independientemente de cual sea la forma explícita de la función de vecindad, esta debe ser tal que $h_{(c,c)}(t) = 1$ para todo t; además para cada t fijo, $h_{ci}(t)$ debe ser creciente en función de $\|r_{c(t)} -\ r_i\|$ y cumplir con $h_{(c,i)}(t) \to 0$ cuando $\|r_{c(t)} -\ r_i\|$ se incrementa.

Una de las definiciones más simple que se encuentran de la función de vecindad es:

$$\begin{aligned} h_{(c,i)}(t) &= 1\ si\ i\ \in\ \eta_c(t) \\ h_{(c,i)}(t) &= 0\ si\ i\ \in\ \eta_c(t) \end{aligned} \qquad (4.5)$$

En este caso $\eta_c(t)$ es una vecindad de $\eta_{c(t)}$ sobre la retícula que se define como:

$$N_c(t) =\ \{i\ \in\ \mathbb{N}\ \|r_{c(t)}\ - r_i\|\ \leq\ \rho(t)\} \qquad (4.6)$$

donde $\rho(t)$ es el radio de la vecindad en el tiempo t.

Otra forma común de la función de vecindad está dada en términos de la función Gaussiana:

$$h_{(c,i)}(t) = \exp\left(\frac{\|r_c - r_i\|^2}{2\rho^2(t)}\right) \tag{4.7}$$

donde $\rho(t)$ corresponde al ancho promedio de $N_c(t)$.

Para efectos de la convergencia del algoritmo, la variación del radio a través del tiempo debe cumplir que $t_i \leq t_j \Rightarrow \rho(t_i) \geq \rho(t_j)$y además $\rho(t) \to 0$ cuando $t \to \alpha$, ver Figura 4.4. Se recomienda que $\rho(1)$ sea más grande que la mitad del diámetro de la red [118].

La función $\alpha(t)$ es el *factor de aprendizaje* y en este caso cumple con la condición $0 < \alpha(t) < 1$ y es no creciente de manera que $\alpha(t) \to 0$ cuando $t \to \alpha$.

Tanto $\alpha(t)$ como $\rho(t)$ son componentes autónomas de la regla de aprendizaje y su principal objetivo es garantizar la convergencia del algoritmo a partir de producir cambios cada vez más locales (centrados en la neurona ganadora) y de menor magnitud.

Durante la evolución del proceso de entrenamiento, los vectores de peso son modificados de manera que cada uno de estos se vuelva representante de una porción en el espacio de entrada. Durante esta evolución es común observar dos etapas: *ordenamiento global* y *refinamiento*.

En proceso de entrenamiento del SOM, hay que ser muy cautelosos al fijar el tamaño inicial a $\rho(0)$. Si desde un comienzo la vecindad es muy pequeña, el mapa no se ordenará globalmente, lo cual implicará que el mapa generado se verá como un mosaico de partes entre las cuales el ordenamiento cambia discontinuamente. Para evitar esto, $\rho(0)$ puede comenzar siendo mayor que el radio de la red.

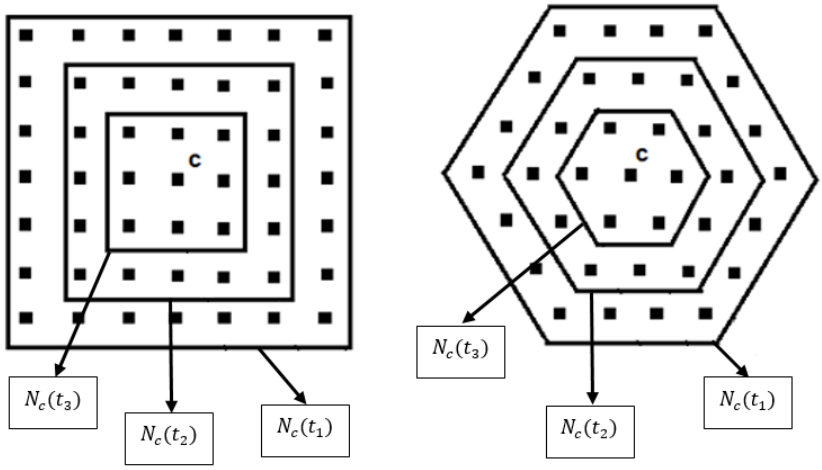


Figura 4.4. Variación en el tiempo del radio de la vecindad [107]

Para el proceso de aprendizaje inicial, se utilizan valores aleatorios para los vectores de referencia $w_i(0)$. En las versiones más simples del SOM los valores sucesivos para los vectores de referencia se determinan recursivamente por el siguiente mapeo de iteraciones:

$$w_i(t+1) = w_i(t) + \alpha(t)h_{ci}(t)\,[x\,(t) - w_i(t)] \qquad (4.8)$$

donde t denota el tiempo. El $x\,(t)$ es un vector de entrada extraído al azar de los datos de entrada en el tiempo t, $h_{ci}(t)$ el núcleo de la vecindad alrededor de la unidad ganadora c y $\alpha(t)$ la tasa de aprendizaje en el tiempo t. El kernel de la vecindad es una función no creciente del tiempo y de la distancia de la unidad i de la unidad ganadora c. Se define así la región de influencia que la muestra de entrada tiene en el SOM.

El entrenamiento se realiza generalmente en dos fases. En la primera fase, se utilizan tasas α_0 relativamente altas de aprendizaje inicial y σ_0 radios de vecindad. En la segunda fase tanto las tasas de aprendizaje como la de los radios de vecindad son pequeños desde el inicio. Este procedimiento corresponde al primer ajuste del SOM aproximadamente al mismo espacio que los datos de entrada y luego un afinamiento detallado del mapa.

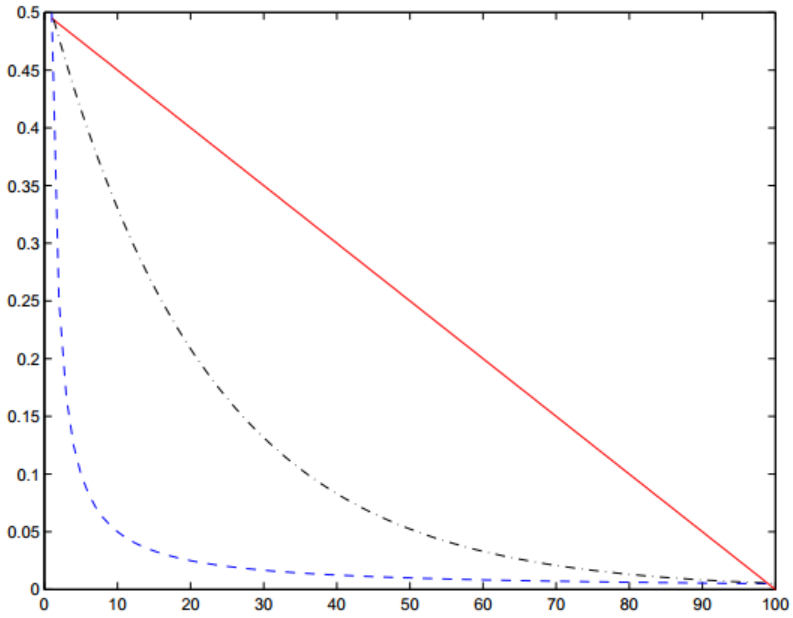


Figura 4.5. Diferentes funciones de la tasa de aprendizaje [121].

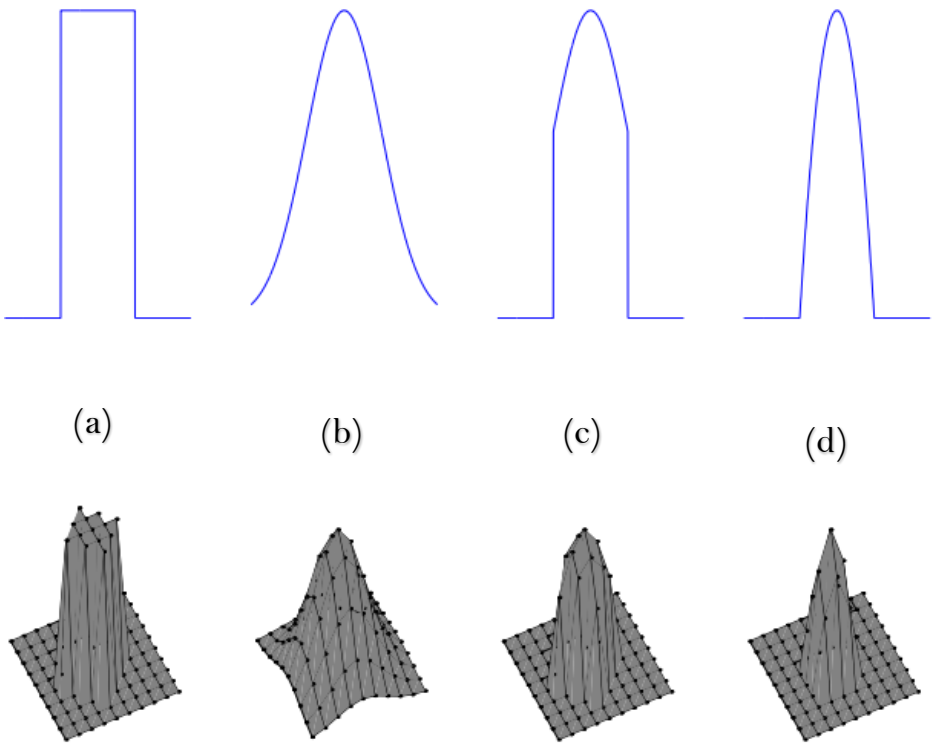


Figura 4.6. Diferentes funciones de vecindad [121].

En la Figura 4.5 se evidencian las diferentes funciones de la tasa de aprendizaje. Estas son:

-***linear*** (línea roja): $\alpha(t) = \alpha_0(1 - t/T)$
-***power*** (línea discontinua negra de puntos) $\alpha(t) = \alpha_0(0.005/\alpha_0)^{t/T}$
-***inv*** (línea azul) $\alpha(t) = \alpha_0(1 + 100\ t/T)$

donde T es la duración del aprendizaje y α_0 es la tasa de aprendizaje inicial.

Según la Figura 4.6, h puede ser expresada de distintas formas, estas son:

(a) ***bubble***: $h_{ci}(t) = 1(\sigma_t - d_{ci})$,
(b) ***gaussian***: $h_{ci}(t) = e^{-d_{ci}^2/2\sigma_t^2}$,
(c) ***cutgauss***: $h_{ci}(t) = e^{-d_{ci}^2/2\sigma_t^2}\ \mathbf{1}(\sigma_t - d_{ci})$,
(d) ***ep***: $h_{ci}(t) = \{0,1 - (\sigma_t - d_{ci})^2\}$

donde σ_t es el radio de la vecindad en el tiempo t, $d_{ci} = \|\boldsymbol{r}_c - \boldsymbol{r}_i\|$ es la distancia entre las unidades del mapa c e i en la cuadricula del mapa, y $\mathbf{1}(\boldsymbol{x})$ es la siguiente función: $\mathbf{1}(\boldsymbol{x}) = 0$ si $x < 0$ y $\mathbf{1}(\boldsymbol{x}) = 1$ si $x \geq 0$. En la parte superior de la Figura 4.6 se muestra la función en una dimensión y la parte inferior en dos dimensiones. El radio de la vecindad usado en Figura 4.6 es $\sigma_t = 2$.

Durante el proceso de entrenamiento se logran apreciar dos etapas, el ordenamiento global y el refinamiento. En la primera de ellas [107], durante los primeros pasos (usualmente los 1000 primeros pasos) se llevan a cabo el ordenamiento de los datos a lo largo y ancho del mapa con el único fin de establecer los pesos de cada neurona y buscar que sean capaces de identificar cierto subconjunto característico dentro del conjunto de datos, y para que las relaciones de cercanía entre las distintas neuronas del mapa reflejen cercanía de los datos correspondientes en el espacio multidimensional del cual provienen. Cuando los pesos se dan por la aleatoriedad de los valores iniciales durante los pasos iniciales se debe tener en cuenta que los valores de $\alpha(t)$ deben ser altos o cercanos a uno, después de hacer la respectiva normalización e ir descendiendo hasta llegar a valores cercanos a 0,2. Cabe decir que la selección optima de estas funciones y sus parámetros se determinar usualmente de forma experimental, ya que no existe algún resultado analítico que garantice dicha selección óptima.

4.5.1.1.1 Algoritmo de entrenamiento secuencial

En este tipo de entrenamiento, el SOM es entrenado de manera iterativa. En cada paso, un vector muestra del conjunto de los datos de entrada es escogido aleatoriamente y la distancia entre él y todos los vectores de pesos del SOM se calculan usando alguna medida de distancia. La neurona cuyo vector de pesos está más cercado al vector de entra da x es llamada la unidad de mejor coincidencia o ***Best-Matching Unit*** (***BMU***), denotada por c en la ecuación:

$$\|x - \omega_c\| = \min_i\{\|x - \omega_i\|\} \tag{4.9}$$

donde $\|.\|$ es una medida de distancia, típicamente es la Distancia Euclidiana.

En este trabajo de investigación se utilizó la *toolbox* de SOM *Matlab*® para el lenguaje de programación. Para este caso puntual, esta distancia es ligeramente más compleja debido a los "***valores ausentes***" los cuales son representados con ***NaN*** en el vector de datos de la matriz. Las componentes que faltan se tratan por exclusión simple luego del cálculo de la distancia, es decir que se supone que su contribución a la distancia $\|x - \omega_i\|$ es cero. Debido a que las variables son las mismas, estas son ignoradas en cada cálculo de la distancia [122]. Otro de los factores que complican un poco el cálculo computacional de la distancia es el "***enmascaramiento***" ya que cada variable es asociada con un factor de ponderación usando una función interna de *Matlab*®. Esto se utiliza principalmente en forma binaria para la exclusión de ciertas variables del proceso de búsqueda de la BMU (1 para ser incluida y 0 para la exclusión). Sin embargo, el enmascaramiento obtiene algunos valores, que pueden ser usados para ponderar variables de acuerdo a su importancia.

Estos dos factores hacen que la medida de distancia usada sea:

$$\|x - w\|^2 = \sum_{k \in K} v_k (x_k - w_k)^2 \tag{4.10}$$

donde K es el conjunto de variable conocidas (no perdidas) del vector de muestra x, x_k y w_k es el k-esimo componente de la muestra y los vectores de peso, y v_k es el k-esimo valor de la máscara.

4.5.1.1.2 Algoritmo de entrenamiento por lotes

Este algoritmo, así como el secuencial también es iterativo, pero en vez de usar un simple vector de datos a la vez, todo el conjunto de datos se presenta al mapa antes de hacer cualquier ajuste, de ahí el nombre de "por lotes". En cada etapa del entrenamiento, el conjunto de datos se divide de acuerdo a las regiones de Voronoi del mapa de vector de pesos. Después, el nuevo vector de pesos se calcula de la siguiente manera:

$$\boldsymbol{\omega}_i(t+1) = \frac{\sum_{j=1}^{n} h_{ic}(t)\boldsymbol{x}_j}{\sum_{j=1}^{n} h_{ic}(t)} \tag{4.11}$$

donde $c = \arg\min_k\{\|x_j - \ \omega_k\ \|\}$ es el índice de las muestras de la ***BMU*** de los datos de muestras x_j. El nuevo vector de pesos es un peso ponderado de los datos de muestra, donde el peso de cada dato de la muestra es el valor de la función de vecindad $h_{ic}(t)$ en su ***BMU*** c. Igual que en el algoritmo de entrenamiento secuencial, los valores perdidos son simplemente ignorados en el cálculo del peso ponderado.

Es de notar que en la versión de aprendizaje por lotes de *k-means*, el nuevo vector de pesos son promedios simples del conjunto de datos de Voronoi. La ecuación anterior será igual a esto, si $h_{ic} = \delta(i, c)$.

De manera alternativa se puede hacer el cálculo de la suma de los vectores en cada conjunto de Voronoi como:

$$\boldsymbol{s}_i(t) = \sum_{j=1}^{n_{V_i}} \boldsymbol{x}_j \tag{4.12}$$

donde n_{V_i} es el número de muestras en el conjunto de Voronoi de la unidad i. Así, los nuevos valores del vector de pesos pueden ser calculados como:

$$\boldsymbol{\omega}_i(t+1) = \frac{\sum_{j=1}^{n} h_{ic}(t)\boldsymbol{s}_i(t)}{\sum_{j=1}^{n} n_{V_i} h_{ic}(t)} \tag{4.13}$$

donde $\boldsymbol{\omega}$ es el número de unidades del mapa.

Retomando las etapas en la ejecución del algoritmo SOM, seguimos ahora con la cuarta, la ***visualización***. El SOM puede representarse de una manera visual, ya que puede contemplarse como una aproximación de la función de densidad de probabilidad de los datos de entrada [107]. La representación *U-Matrix* (*Unified Distance Matrix*) del SOM (Figura 4.7) visualiza la distancia entre neuronas adyacentes. Esta distancia se representa con diferentes colores entre los nodos adyacentes. Un color oscuro representa una distancia grande mientras que uno de baja intensidad significa que los patrones están cerca unos de otros. Lo anterior nos dice que en los resultados obtenidos en la visualización se pueda hablar de "clases" o áreas claras, y "separadores" o áreas oscuras. Esta forma de representación es de gran ayuda para "visualizar" los datos de entrada sin tener información a priori sobre las clases.

Por último, encontramos la etapa de ***validación*** donde se prueba el modelo a utilizar para tener la certeza de que los datos de salida son razonables y ciertos. En la mayoría de los casos esta etapa se hace usando un conjunto independiente de datos muy parecido al del entrenamiento, pero sin ser parte de él, siendo esta porción como un caso representativo del caso general.

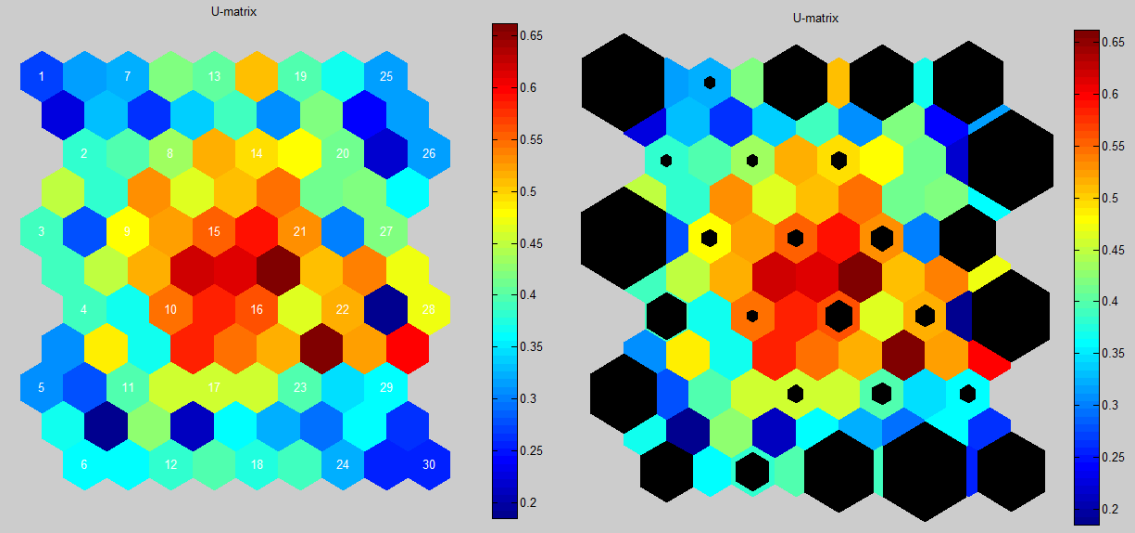


Figura 4.7. Matriz de distancia Unificada (U-matrix, IZQ) y densidad de los clusters (U-matrix, DER). [123]

4.6 Conclusiones

A lo largo del capítulo hemos introducido los principios básicos de RNAs, así como las arquitecturas y el aprendizaje de las neuronas. Se hace un estudio de las Redes Neuronales Autoorganizables, detallando las redes

SOM. Las SOM se muestran como un posible método de reducción de dimensión. Sin embargo, el uso de SOM presenta algunas dificultades, sobre todo cuando la dimensión del espacio de características (entrada) es muy alta. Por otra parte, las dimensiones y el tamaño del SOM tiene que ser establecido antes del proceso de entrenamiento, y sus valores óptimos tienen que determinarse mediante ensayo y error. Sin embargo, el número de neuronas en el espacio de salida determina la calidad del mapa y, a continuación, el rendimiento del proceso de clasificación (es decir, las clases deben presentar suficientes diferencias entre ellas y estar lo suficientemente alejadas del resto de clases en el espacio de salida).

No obstante, la importancia de RNAs radica en problemas de tipo no lineal. Por su naturaleza experimental, su base teórica es en su gran mayoría heurística, y su implementación se sustenta principalmente en métodos basados en la experiencia. Sin embargo, esta rama de investigación ha sido y es objeto de un trabajo considerable, y los resultados obtenidos hasta el momento son prometedores para el futuro de RNAs.

Hemos realizado una extensa investigación de las RNA, incluyendo este estudio en esta tesis como una revisión de trabajos previos realizados en esta área. Esta exploración se hace justo antes de explicar las redes SOM y el funcionamiento de las mismas, siendo estas, parte fundamental de uno de los pilares en la elaboración de esta tesis.

Parte II

Análisis en componentes de conexiones

Capítulo 5. Procedimiento basado en mapas autoorganizativos probabilísticos

En este capítulo se muestra el método de selección de características y clasificación que se propone para la detección de anomalías. La descripción del método se ha dividido en tres secciones. En primer lugar, la generación de características y filtrado PCA (Sección 5.1), donde se muestra como se generaron las características para este estudio y se detalla el proceso de filtrado PCA. En la Sección 5.2 se utiliza la tasa discriminante de Fisher (FDR) para la selección de autovectores (*eigenvectors*), dando una solución al problema del ruido en el conjunto de datos y en el de los valores únicos que no son lo suficientemente discriminantes, con información redundante, o no pertinente, y que puede resolverse mediante la selección de los componentes principales de acuerdo con el criterio FDR. Además, se ha considerado un hibrido entre las *Bayesian SOM* y específicamente un GMM para entrenar el mapa solo una vez, mientras que los resultados de la clasificación se puedan lograr con la modificación de las probabilidades de activación previa de las unidades SOM, de tal manera que permitan que el nivel de activación que se utiliza para reconocer los patrones correspondientes asociados a las conexiones normales y anómalas de red en este caso puedan ser identificados. Para completar la descripción de nuestro método, en capítulo siguiente, se muestran los experimentos y resultados obtenidos utilizando como clasificador un SOM bayesiano, que se ha implementado a través del modelado del mapa mediante mezclas de distribuciones Gaussianas.

La mayor parte de los procesos biológicos, como por ejemplo la actividad cerebral, exhiben localidad: las características que contienen información de regiones anatómicas que están "próximas", tienen probabilidad de estar altamente correlacionadas. En nuestro caso, la detección efectiva de las conexiones anómalas se basa en la disimilitud de distancias en un SOM del proceso de clasificación del total de conexiones. La localidad es un término altamente estudiado y muy notorio cuando nos referimos a procesos biológicos, siendo el ejemplo más usado para representar la actividad cerebral en inteligencia artificial. En ésta, la proximidad de características implica inherentemente una alta probabilidad de correlación. Para la

investigación efectuada, la detección de conexiones anómalas y normales en tráficos de red, implica identificación de las unidades que corresponden a las situaciones anómalas.

Las técnicas basadas en estadística univariante, debido a la gran cantidad de características que se manejan en el proceso de clasificación de este trabajo de investigación, son indicadas para realizar un proceso eficiente de clasificación de los paquetes recibidos, ya que este tipo de técnicas no contribuyen eficazmente en la recuperación de la información que reside en la correlación de las características. Por lo tanto, el uso de técnicas estadísticas multivariantes aplicadas al diagnóstico de conexiones anómalas está plenamente justificado para superar las limitaciones impuestas por las aproximaciones univariantes [124, 125].

El enfoque que se propone está basado en la descomposición o filtrado del conjunto de datos para extraer las características más relevantes del mismo.

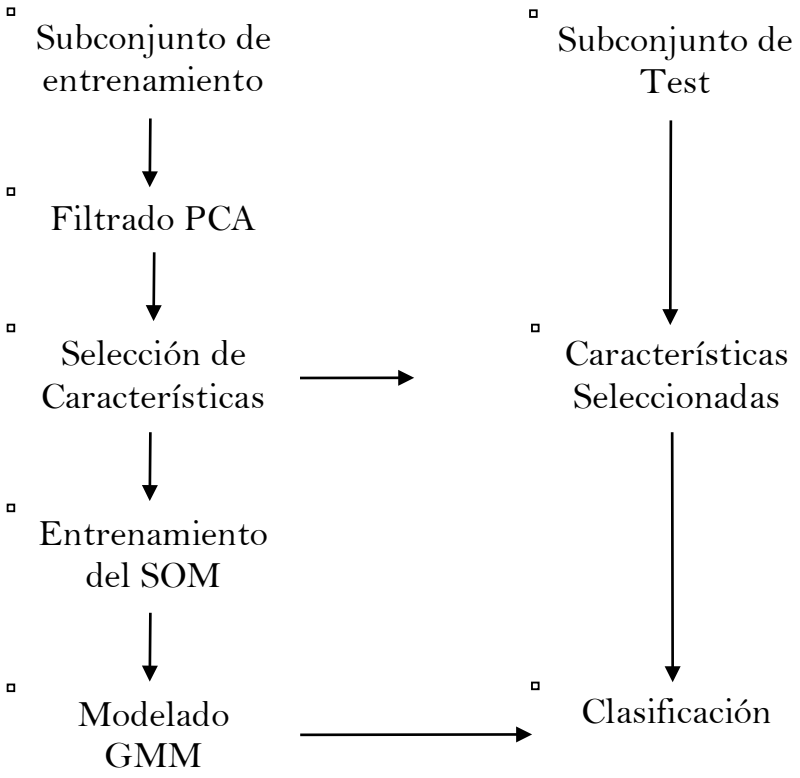


Figura 5.1. Diagrama de bloques del sistema de detección de anomalías propuesta

La figura 5.1 muestra el diagrama de bloques correspondiente al procedimiento planteado. En esta figura muestra el uso de los subconjuntos

de entrenamiento y test o prueba. En una primera instancia, el subconjunto de entrenamiento es sometido a un proceso de filtrado usando la técnica Análisis de Componentes Principales (PCA), después de ello, se elimina el "ruido" que pueda tener el conjunto de datos resultante y se seleccionan las características más relevantes mediante el Factor Discriminante de Fisher (FDR). Con estas características, usamos los Mapas Autoorganizativos (SOM) para hacer el proceso de entrenamiento, etiquetando los datos a partir de d prototipos N-dimensionales, o vectores modelo. Este modelo de aprendizaje activa una unidad específica (es decir, el correspondiente BMU) cada vez que una nueva instancia de datos se presenta a la SOM. Habiendo entrenado el SOM, se hace una interpretación probabilística de la SOM mediante el modelado de los prototipos usando un Modelo de Mezcla Gaussiana (GMM).
De manera paralela, al momento de hacer el proceso de clasificación, se usa el subconjunto de datos de test, en el que se le seleccionan las características más representativas usando (FDR), y usando el clasificador ya entrenado, se lleva a cabo la clasificación.

5.1 Generación de características y filtrado PCA

Como se ha indicado en capítulo 3, *PCA* es una técnica estándar para extraer las características más significativas de un conjunto de datos. Se basa en la acción de una transformación lineal (también conocida como la trasformación de *Karhunen-Loeve*) [126] sobre un conjunto de datos de media nula, que diagonaliza su matriz de covarianza. Matemáticamente se define como una transformación lineal ortogonal que transforma los datos en un nuevo conjunto de variables que agrupan la mayor cantidad de varianza, denominadas componentes principales (*Principal Component - PC*). La primera componente principal contendrá las características de los datos con mayor contribución a la varianza, seguida por orden decreciente en su valor de la varianza por la segunda componente principal, tercera, etc. Existen varias construcciones equivalentes entre ellas, que conducen a la obtención de estas componentes principales, siendo cada una más apropiada en un contexto diferente.

La selección de características es un paso clave en problemas de clasificación, ya que contribuye a la eliminación de la entrada redundante o irrelevante características no solo para reducir los tiempos de computación para el aprendizaje, sino también para mejorar la precisión del clasificador [127].

El PCA ha sido ampliamente utilizado en muchas aplicaciones para la extracción de información más relevante en conjuntos de datos. De hecho,

se ha utilizado con éxito en aplicaciones de reconocimiento facial [128]. En este caso, PCA se utiliza para obtener un nuevo conjunto de características no correlacionadas a partir de un conjunto correlacionados. Por lo tanto, PCA genera un conjunto de vectores ortogonales base de modo que los datos se pueden expresar como una combinación lineal de los vectores de esa base.

El procedimiento se describe en lo que sigue. Sea $X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_{N_t}\}$, $\boldsymbol{x_i} = (x_i^1, \dots, x_i^n)^T$ son las muestras de datos de entrada (muestras de entrenamiento). Una versión una versión de media cero de los datos se puede obtener restando la media $(\bar{X})$ a cada uno de ellos, $Y = X - \bar{X}$, donde $y_i \in \mathbb{R}^n$, $y_i = (y_i^1, \dots, y_i^n)^T$, $i = 1, \dots, N_t$. El PCA construye los vectores $\boldsymbol{u_k} = (u_k^1, \dots, u_k^n), k = 1, \dots, N_t$ como:

$$\lambda_k = \frac{1}{M} \sum_{r=1}^{N_t} (\boldsymbol{u}_k^T \boldsymbol{y}_r)^2 \tag{5.1}$$

El vector $\boldsymbol{u}_k, k = 1, \dots, N_t$ verifica que $\boldsymbol{u}_l^T \boldsymbol{u_k} = \delta_{\boldsymbol{lk}}$ ($\delta_{\boldsymbol{lk}}$ es la delta de Kronecker). Los vectores $\boldsymbol{u}_k$ y los escalares λ_k son los vectores propios y valores propios, respectivamente, de la matriz de covarianza calculada como $C = YY^T$. Es importante resaltar la diferencia entre el método presentado y el de *eigenconnections* propuesto [129, 130], e inspirado en el método de reconocimiento facial utilizando vectores propios [128]. En el caso propuesto en este trabajo, se utilizan vectores propios para generar un nuevo espacio de características que nos permite eliminar el ruido que comprende la información discriminante en un número reducido de características. Así, las muestras de los datos de entrenamiento se proyectan en el espacio abarcado por los vectores propios para generar el conjunto de características no correlacionadas que mejor describan el conjunto de datos. Estas características se utilizan, además, para entrenar el clasificador basado en SOM descrito en la Sección 4.5 Para clasificar una nueva instancia de datos **v**, esta tiene que ser proyectada en el espacio de autovalores, obteniéndose su correspondiente vector de características:

$$\boldsymbol{\omega}_k = (\boldsymbol{v} - \bar{X}) * \boldsymbol{u}_k \tag{5.2}$$

Por otro lado, los datos originales pueden ser reconstruidos a partir de las componentes principales como:

$$\boldsymbol{v}_{\boldsymbol{k}}^{\boldsymbol{rec}} = \bar{X} + \boldsymbol{\omega}_k * \boldsymbol{u}_k^T \tag{5.3}$$

donde $\boldsymbol{v}_{\boldsymbol{k}}^{\boldsymbol{rec}}$ es la reconstrucción de $\boldsymbol{v}$ usando el autovector k.

Aunque en el problema considerado en este trabajo, las dos primeras componentes principales representan más del 95 por ciento de la varianza, esto no garantiza la capacidad discriminante de las proyecciones. Por lo tanto, los vectores propios seleccionados están ordenados por su poder discriminante de acuerdo con el valor FDR, tal y como se explica en la siguiente sección.

5.2 Selección de componentes mediante el factor discriminante de Fisher (FDR)

En muchos problemas (como es el caso de la clasificación de anomalías en el conjunto *NSL-KDD*), el conjunto de datos es ruidoso y contiene columnas con valores únicos o casi únicos. Por otra parte, existen otras columnas que no contienen valores únicos lo suficientemente discriminantes, contienen información redundante, o no pertinente, y que dan lugar a errores de clasificación. Por esta razón, se hace necesaria una etapa de pre-procesamiento para filtrar los datos [131, 132].
Aunque el PCA ordena los principales componentes por su varianza asociada (autovalor o valor propio), esto no garantiza que los vectores propios con los valores propios más altos son los más discriminantes. Este problema puede resolverse mediante las selecciones de las componentes principales de acuerdo con el criterio FDR, a partir del siguiente método que proponemos:

Sea $H = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_{N_t}\}$ el conjunto las muestras de entrenamiento, sean E sus versiones desplazadas, y $V = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_n\}$ los autovectores de la matriz H. Un nuevo conjunto de datos de entrenamiento se puede derivar de H proyectándola en cada autovector. Por lo tanto, se define:

$$\boldsymbol{\xi} = v_i^T * H \tag{5.4}$$

Cada columna $\boldsymbol{\xi}_i$ en $\boldsymbol{\xi} = \{\boldsymbol{\xi}_i, \dots, \boldsymbol{\xi}_n\}$ corresponde a la proyección de H en el autovector $\boldsymbol{v}_i$. Como el PCA ordena los autovectores de acuerdo con un orden de varianza decreciente, los primeros autovectores comprenden la mayor parte de la varianza. Sin embargo, esto no es suficiente para asegurar una buena capacidad de separación de clases. De esta manera, los autovector

se clasifican de acuerdo a su valor FDR:

$$FDR = \sum_{i}^{M} \sum_{j \neq i}^{M} \frac{(v_i - v)^2}{\sigma_i^2 + \sigma_j^2} \tag{5.5}$$

Para el caso multiclase, donde σ_i y v_i son la varianza y la media de la clase i, respectivamente, la ecuación (5.4) se puede transformar en:

$$FDR = \frac{\left(v_i - v_j\right)^2}{\sigma_i^2 + \sigma_j^2} \tag{5.6}$$

para el caso de separación de 2 clases.

En este trabajo de investigación se propuso eliminar el ruido del conjunto de datos al utilizar únicamente los autovectores con la capacidad de separación máxima de la clase. Esto se logra mediante la selección de las proyecciones que proporcionen los r valores más bajos de FDR, y restando la reconstrucción de las muestras que utilizan estos autovectores del conjunto de datos original, como se indica en la siguiente ecuación:

$$\widehat{H} = H - \sum_{i=1}^{r} \boldsymbol{v}_i \boldsymbol{\xi}_i \tag{5.7}$$

El algoritmo completo de selección de características descrito, que utiliza FDR para determinar aquellos vectores propios de acuerdo a su poder discriminante se muestra en el pseudocódigo 5.1:

(1) Restar la media de las muestras de entrenamiento: $E = H - \overline{H}$.
(2) Calcular las componentes principales v_i^T, ver ecuación (5.20).
(3) Calcular la proyección $\boldsymbol{\xi}_i$ de las muestras de entrenamiento en los *eigenvectors* de acuerdo con la ecuación (5.4).
(4) Ordenar los autovectores de acuerdo con el poder discriminante de las proyecciones en función de su valor FDR
(5) Restar la proyección de las muestras de entrenamiento en los autovectores con valores inferiores FDR en el conjunto de datos tal como se indica en la ecuación (5.7).

Pseudocodigo 6.1

Una variante de este método ha sido utilizada en [133, 134]. El filtrado del

conjunto de datos $\hat{X}$ contiene el mismo número de características que X, pero su ruido corresponde a los autovectores menos discriminantes. Posteriormente, el conjunto de datos filtrado se proyecta en los k autovectores con el valor más alto de FDR (es decir, los más discriminantes), reduciendo así la dimensión del conjunto de datos a k. De esta manera, el método descrito permite la selección de los k autovectores más discriminantes y, posteriormente la generación de características discriminantes mediante la proyección de los datos originales sobre esos k autovectores.

5.3 Clasificación usando *Bayesian SOM*

Los SOM [107] son uno de los modelos de redes neuronales más populares para el aprendizaje no supervisado. Permiten agrupar datos del conjunto de entrada a partir del estado del mapa de salida del SOM una vez entrenado. Algunas características a tener en cuenta en los SOM desde el punto de vista del problema que pretendemos resolver son:

1. *Modelado espacio de entrada*. Los prototipos $\boldsymbol{\omega}_i$ (donde (i) se refiere al índice de la unidad en el mapa) calculados durante el entrenamiento SOM proporcionan una aproximación del espacio de entrada.
2. *Orden topológico*. Las unidades en el mapa de salida están organizadas en mallas 2D o 3D, y su posición depende de las características específicas del espacio de entrada. De esta manera, el índice de una unidad podría expresarse como tuplas (i_1, i_2) or (i_1, i_2, i_3) , respectivamente, para entramados 2D o 3D.
3. *La selección de características*. El algoritmo SOM produce una serie de prototipos desde el espacio de datos de entrada. Por lo tanto, el algoritmo no solo reduce la dimensión, sino también el tamaño espacio de entrada, tal como está representado por los vectores prototipos.

El algoritmo de SOM se explicó de manera genérica en la Sección 5.5.1.1. y se mostró como el mapa SOM se compone de d unidades, cada una representada por un vector modelo n-dimensional $\boldsymbol{\omega}_i$. Para cada instancia de datos de entrada $\boldsymbol{v}$, la BMU se define como la unidad $\boldsymbol{\omega}_i$ más cercana a $\boldsymbol{v}$, es decir:

$$\|\boldsymbol{\omega}_i - \boldsymbol{v}\| \leq \|\boldsymbol{\omega}_j - \boldsymbol{v}\|, \qquad \forall \boldsymbol{v} \in X, i \neq j$$

donde, $\|.\|$ es la distancia euclidiana y X en el conjunto de datos de entrenamiento. Una vez que se determina la BMU para la actual iteración, los vectores modelo se actualizan de acuerdo con la regla:

$$\boldsymbol{\omega}_i(l+1) = \boldsymbol{\omega}_i(l) + \alpha(l)h_i(l)(\boldsymbol{v} - \boldsymbol{\omega}_i(l))$$

donde $\alpha(l)$ es la tasa de aprendizaje y $h_i(l)$ la función que define la vecindad alrededor de la BMU $\boldsymbol{\omega}_i$. Usualmente[2], $\alpha(l)$ disminuye siguiendo una regla de decrecimiento exponencial [107] y h_i es una Gaussiana [107] cuya anchura se reduce en el tiempo (a medida que crece el número de iteraciones).

5.3.1 Mapas Autoorganizativos Probabilísticos (PSOM)

Los PSOM (*Probabilistic Self-Organizing Map*) son también llamados "PbSOM" [135]. El modelo considerado en esta tesis [136], tiene una función de probabilidad definida en el espacio de entrada. Esta función está definida como una mezcla de probabilidades donde cada unidad del mapa está asociada a un componente de la mezcla, es decir, sea D la dimensión del espacio observado (entrada), la probabilidad de los datos observados $\boldsymbol{t} \in \Re^D$ esta dada por la mezcla de la función de probabilidad del mapa así:

$$p(\boldsymbol{t}) = \sum_{i=1}^{N} \pi_i p(\boldsymbol{t} \mid i) \qquad (5.8)$$

donde N es el número de componentes de la mezcla (unidades), π_i, es la probabilidad a priori o proporción de mezcla de la unidad i, y $p(\boldsymbol{t} \mid i)$ es la función de densidad de probabilidad asociada a i.

El mapa estará definido entonces entre el espacio latente $\Re^2$ y el espacio de entrada $\Re^D$ así:

$$\boldsymbol{t} = \boldsymbol{W}\boldsymbol{\Phi}(\mathbf{u}) \qquad (5.9)$$

donde $\boldsymbol{\Phi}$ es un conjunto de funciones de base constante B, $\boldsymbol{\Phi}(\mathbf{u})$ es un vector columna $B \times 1$, y $\boldsymbol{W}$ es la matriz mapeada $B \times D$ a ser ajustada. Una distribución de probabilidad discreta se define sobre una malla regular de

[2] Se debe tener en cuenta que *i* es un índice lineal que identifica los vectores prototipo

puntos latentes $\mathbf{u}_i \in \Re^2$ así:

$$P(\mathbf{u}) = \frac{1}{N}\sum_{i=1}^{N} \delta(\mathbf{u} - \mathbf{u}_i) \tag{5.10}$$

donde δ es la función delta de Dirac [137]. Entonces, el modelo particular dado debe definir la función de densidad $p(\boldsymbol{t} \mid \mathbf{u}, \boldsymbol{W})$ de los puntos del espacio de entrada $\boldsymbol{t}$ dado un punto latente $\mathbf{u}$ y una matriz de mapeo $\boldsymbol{W}$.

Por el contrario, los modelos que no utilizan un espacio latente deben diseñar un algoritmo de entrenamiento, que asegure que los parámetros de mezclas de unidades vecinas i, j en el mapa son similares, de manera que

$$p(\boldsymbol{t} \mid i) \approx p(\boldsymbol{t} \mid j) \tag{5.11}$$

En otras palabras, la topología del mapa guía el proceso de aprendizaje con el fin de lograr lo planteado en 5.11. Para cada muestra de entrenamiento $\boldsymbol{t}_n$, es escogida una unidad ganadora $Win(n)$ y sus parámetros son ajustados a la muestra. Esta unidad ganadora es usualmente definida como la de mayor probabilidad a posteriori ha sido generada por las muestras de entrenamiento, así:

$$\begin{aligned} Win(n) &= \arg\max_i \{P(i \mid \boldsymbol{t}_n)\} \\ &= \arg\max_i \{\pi_i p(\boldsymbol{t}_n \mid i)\} \end{aligned} \tag{5.12}$$

Sus vecinos i también se ajustan, pero con una tasa de aprendizaje η_i que decae con la distancia topológica $d\,(i,k)$ para el ganador de la siguiente manera:

$$d(i, Win(n)) \geq d(j, Win(n)) \Leftrightarrow \eta_i \leq \eta_j \tag{5.13}$$

Muchas de las propuestas utilizan una función de vecindad Gaussiana Λ, esta función no debe ser confundida con la función de densidad de probabilidad $p(\boldsymbol{t} \mid i)$ de los componentes de la muestra que fue definida anteriormente, que varía con el paso de tiempo n y depende de un radio de vecindad decreciente $\Delta(n)$ así:

$$\eta_i(n)\, \alpha\, \Lambda\left(i, Win(n)\right) =$$

$$\exp\left(-\left(\frac{d\big(i, Win(n)\big)}{\Delta(n)}\right)^2\right)$$

$$\Delta(n+1) \leq \Delta(n)$$

Se han propuesto dos enfoques principales de razonamiento para justificar la condición 5.13. La primera de ellas sigue la propuesta original de Kohonen [77], por lo que la restricción de autoorganización no está relacionada con ninguna probabilidad, pero se introduce como un medio para reforzar el aprendizaje de las unidades vecinas. Otro enfoque, propuesto por Heskes [138], interpreta la función de vecindad como una probabilidad de confusión, es decir, la probabilidad de que la unidad ganadora es k, mientras que la muestra de entrada fue de hecho generada por la mezcla de componente i como sigue:

$$\Lambda\,(i,k) = P(i,|k = Win(n)) \tag{5.14}$$

Se debe tener en cuenta que en este caso se debe normalizar la función de vecindad dividiendo por una constante adecuada, de manera que se puede interpretar como una probabilidad adecuada así:

$$\sum_{i=1}^{H} \Lambda\,(i,k) = 1 \tag{5.15}$$

El enfoque de Heskes está más de acuerdo con un modelo probabilístico, ya que proporciona una clara interpretación probabilística de la restricción de autoorganización.

La *expectativa de maximización* [139 - 142], asume la existencia de datos ocultos, $\boldsymbol{\tau}$, que junto con los datos observados, $\boldsymbol{t}$, forman los datos completos de entrada $(\boldsymbol{t}, \boldsymbol{\tau})$. Seguidamente, se sigue un procedimiento iterativo, con dos pasos que se ejecutan en la siguiente secuencia.

(1) El paso E (*Expectativa*) consiste en calcular la expectativa de probabilidad del logaritmo de los datos completos, dados los valores actuales de los parámetros del vector de pesos $\boldsymbol{\Theta}(n)$ [135], los datos observados $\boldsymbol{t}$, y los nuevos valores tentativos de los parámetros $\boldsymbol{\Theta}$ así:

$$\zeta = \left(\boldsymbol{\Theta}, \boldsymbol{\Theta}(n)\right) = E_{\boldsymbol{\tau}}[\log p(\boldsymbol{t}, \boldsymbol{\tau} \mid \boldsymbol{\Theta}) \mid \boldsymbol{t}, \boldsymbol{\Theta}(n)] \quad (5.16)$$

(2) El paso M (*Maximización*) obtiene los nuevos parámetros $\boldsymbol{\Theta}(n+1)$ mediante la maximización de la probabilidad esperada ζ con respecto a $\boldsymbol{\Theta}$ así:

$$\boldsymbol{\Theta}(n+1) = \arg\max_{\boldsymbol{\Theta}} \zeta\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}(n)\right) \quad (5.17)$$

Este procedimiento se lleva a cabo por lo general en el modo por lotes como se explicó en la Sección 4.5.1.1.2, es decir, todos los datos de entrenamiento disponibles son considerados a la vez.

5.3.2 Inicialización del SOM

Para las implementaciones realizadas en este trabajo, el SOM se ha inicializado linealmente a fin de evitar efectos aleatorios tal como se muestra en [107]. La inicialización del prototipo lineal de SOM tiene como objetivo acomodar los autovalores y autovectores de los datos de entrenamiento [100, 107, 143].

Este método de inicialización implica que la primera dimensión de los prototipos se asigna proporcionalmente a la primera componente principal y que la segunda dimensión se asigna proporcionalmente a la segunda componente principal. Por ejemplo, en mapas 2D, los valores iniciales del vector prototipo con coordenadas (i_1, i_2) en el mapa de salida se determinaron mediante la ecuación (5.18), que define el centroide del subespacio a lo largo de los dos componentes de datos de entrada principales[3]:

$$\boldsymbol{\omega}_i = \bar{X} + \frac{\sigma_1}{nd_1}\left(\boldsymbol{pc}_1\left(i_1 - \frac{nd_1}{2}\right) + \boldsymbol{pc}_2\left(i_2 - \frac{nd_2}{2}\right)\right) \quad (5.18)$$

En la ecuación (5.18) $\bar{X}$ es el vector medio de los datos de entrada a lo largo de cada dimensión, σ_1 es la desviación estándar de la primera componente principal, nd_1 y nd_2 son el número de unidades en la primera y segunda dimensiones mapa, respectivamente, y $\boldsymbol{pc}_1$ y $\boldsymbol{pc}_2$ están en la primera y

[3] Para este caso (i_1, i_2) es el subíndice que identifica el vector prototipo con índice *i* en el mapa SOM.

segunda componente principal de los datos de entrada, respectivamente.

Una vez el mapa ya está entrenado, cada prototipo representa un conjunto de vectores de entrada (los vectores de entrada que activan este vector como su BMU). En otras palabras, el SOM cuantifica el conjunto de datos en d prototipos N-dimensionales, o vectores modelo. Este modelo de aprendizaje activa una unidad específica (es decir, el correspondiente BMU) cada vez que una nueva instancia de datos se presenta a la SOM.

5.3.3 Modelado del SOM mediante Mezclas Gaussianas (GMM)

En este Tesis, se plantea el modelado del SOM mediante mezclas de distribuciones Gaussianas (GMM), para cuantificar la respuesta de cada unidad, proporcionando patrones de activación difusos en lugar de binarios. Es posible medir la respuesta de las unidades del mapa, en lugar de calcular la BMU como la unidad más cercana a los datos de entrada correspondientes. De esta manera, una interpretación probabilística de la SOM podría obtenerse mediante el modelado de los prototipos usando un modelo de mezcla gaussiana (GMM) [100]. El objetivo principal que existe detrás del uso de un GMM es entrenar el mapa solo una vez, mientras que un ajuste posterior de la respuesta del mapa (es decir, los resultados de la clasificación) se puedan lograr mediante la modificación de las probabilidades de activación previa de las unidades SOM, de tal manera que permitan que el nivel de activación se puede utilizar para reconocer los patrones correspondientes, asociados a las conexiones de red normales o anómalas, en cada caso.

Por lo tanto, las anomalías se pueden detectar ya que se desvían de los patrones de activación normales [144]. En este SOM probabilístico, la BMU se determina no solo mediante el cálculo de la distancia mínima a un vector de entrada dado, sino también teniendo en cuenta la probabilidad de una unidad de ser la BMU. La probabilidad previa de cada unidad de mapa i puede ser calculada experimentalmente, teniendo en cuenta la probabilidad de activación del conjunto de entrenamiento de manera similar a [100], y como se muestra en la siguiente ecuación:

$$P_i = \frac{\#\tilde{X}_i}{\#\tilde{X}} \tag{5.19}$$

donde $\#\tilde{X}$ es el número de vectores de entrada y $\#\tilde{X}_i$ es el número de vectores cuyo prototipo más cercano es $\boldsymbol{\omega}_i$, tal como se define en la ecuación (5.20) (conjunto de Voronoi de la unidad i, o campo receptivo de la unidad i) [145]:

$$\tilde{X}_i = \{\boldsymbol{x} \in X / \ \|\boldsymbol{x} - \boldsymbol{\omega}_i\| \leq \|\boldsymbol{x} - \boldsymbol{\omega}_k\| \ \ k = 1, \ldots N, i \neq k\} \tag{5.20}$$

donde X es el conjunto de patrones de entrenamiento.

El GMM se construye de acuerdo con la ecuación. (5.21) usando N componentes (una para cada prototipo SOM), donde el peso P_i para cada Gaussiana corresponde a las probabilidades previas calculadas en la ecuación (5.19):

$$p(\boldsymbol{x}) = \sum_{i=1}^{N} p_i P_i(\boldsymbol{x}) \tag{5.21}$$

En la ecuación (5.21), $P_i(\boldsymbol{x})$ es una distribución Gaussiana n-dimensional [100, 144] que sirve como vector prototipo y se calcula así:

$$P_i(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^n |\sum_i|^{1/2}}} \, e^{\left(-(1/2)(x-\mu_i)^T \sum_i^{-1}(x-\mu_i)\right)} \tag{5.22}$$

La media de cada componente gaussiano individual, μ_i, es el vector prototipo de su correspondiente unidad, mientras que la matriz de covarianza $\sum_i$ para el i-componente viene dado por la dispersión de las muestras de datos alrededor del vector modelo $\boldsymbol{\omega}_i$. Una vez que el modelo GMM se ha construido, la respuesta de la unidad k para una entrada dada $\mathbf{x}$ puede calcularse como la probabilidad posterior utilizando el teorema de Bayes:

$$P(\boldsymbol{\omega}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\boldsymbol{\omega}_k)p(\boldsymbol{\omega}_k)}{p(\mathbf{x})} \tag{5.23}$$

En consecuencia, en la ecuación (5.23), $P_i(\boldsymbol{\omega}_k|\mathbf{x})$ representa la probabilidad de que un vector de muestra $\mathbf{x}$ pertenece a la clase $\boldsymbol{\omega}_k$ (probabilidad posterior), $p(\mathbf{x}|\boldsymbol{\omega}_k)$ es la función de densidad de probabilidad del prototipo

$\boldsymbol{\omega}_k$ calculada a partir de la GMM según la ecuación (5.21) (eso es, $P_k(\mathbf{x})$), y $p(\mathbf{x})$ es una constante de normalización que hace que la densidad de probabilidad posterior se iguale a uno y, por lo tanto, se puede calcular como:

$$p(\mathbf{x}) = \sum_{k=1}^{NC} p(\mathbf{x}|\boldsymbol{\omega}_k)p(\boldsymbol{\omega}_k) \tag{5.24}$$

donde NC es el número de clases.

La Figura 5.2. muestra las probabilidades apriori de activación, p_i, para cada unidad del mapa. Además, se hace una clasificación binaria donde las conexiones normales se asignan en unidades representadas como círculos, mientras que anómalas en la red se asignan en unidades representados como cuadrados.

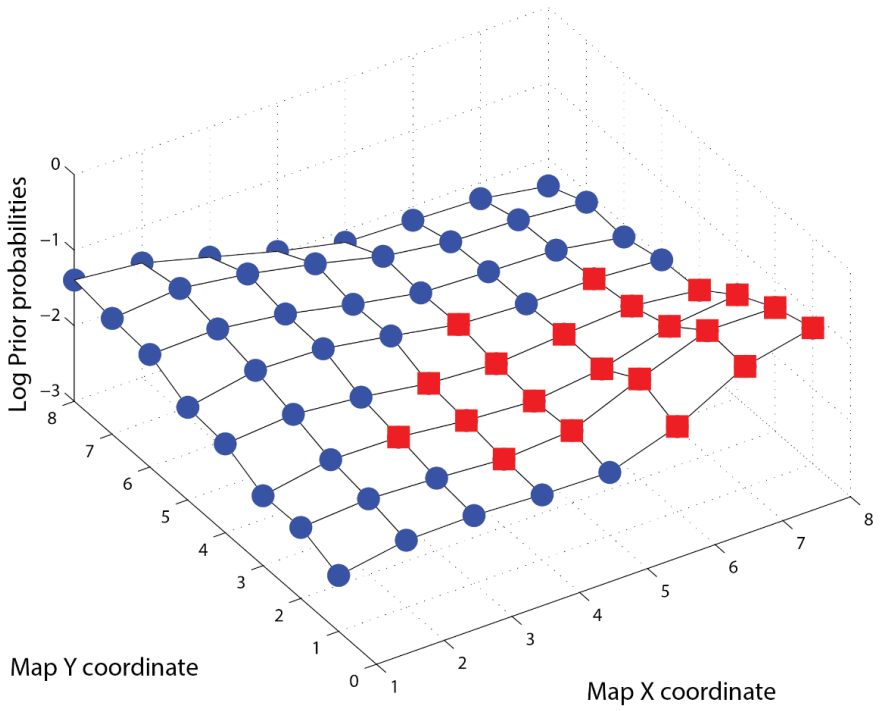


Figura. 5.2. SOM de activación de las probabilidades logarítmicas a priori para cada unidad.

La probabilidad a posteriori se puede utilizar para clasificar nuevas muestras. En este trabajo, las probabilidades a posteriori se han utilizado para reetiquetar las unidades que permanecen sin etiqueta durante el proceso de entrenamiento del SOM. En la Figura 5.2, se muestra el SOM de activación de las probabilidades logarítmicas apriori para cada unidad según la ecuación (5.19), es decir, log $(\#\tilde{X}_i/\#\tilde{X})$. Los círculos y cuadrados

representan la etiqueta de cada unidad, correspondiéndose a las conexiones de redes normales y anómalas, respectivamente. La altura de las unidades (eje z) representa la probabilidad apriori de activación con patrones normales (azul) y atacantes (rojo).

Para el escenario de la Figura 5.2, se puede ver que las muestras normales activan la mayoría de las unidades normales (círculos azules), manteniendo dicha mayoría sobre las unidades que representa anomalías (cuadrados rojos).

5.4 Conclusiones

En este capítulo se describen las principales contribuciones de esta investigación. se propone el uso de un procedimiento basado en mapas auto organizativos probabilísticos PSOM, basándose en la propuesta previa de mapas autoorganizativos que se ha descrito, y en un nuevo enfoque para el filtrado y la selección de características basada en PCA y FDR respectivamente. Dicho procedimiento después de hacer las respectivas implementaciones de los métodos de filtrado y la selección muestran un comportamiento general satisfactorio, con una precisión de clasificación del 90%.

A continuación, se analizarán experimentalmente las prestaciones de este procedimiento en el problema de detección de intrusos.

Capítulo 6. Configuración experimental

En este capítulo se describe la base de datos sobre la cual se realizará el estudio de este trabajo de investigación. Además, se detallan otras bases de datos disponibles para la realización de métodos similares como lo es la KDD cup'99. La base de datos se detalla en su totalidad desde sus atributos hasta sus características. Se hace un análisis del procesado de los datos, así como del proceso de normalización del conjunto de datos. En el siguiente capítulo se evaluarán las prestaciones del procedimiento que se propone en esta tesis para la detección de intrusos a partir de los mapas autoorganizativos y el uso de PCA y FDR.

6.1 El conjunto de datos

El conjunto de datos utilizado en los trabajos realizados para la consecución de los resultados de este documento es el NSL-KDD [146]. Este conjunto surge como una mejora de los datos del concurso KDD cup´99 [147]. En este último conjunto de datos se incluye una versión reducida de la amplia variedad de intrusiones militares simuladas en un entorno de red, proporcionadas por DARPA *Intrusion Detection Program Evaluation* en el año 1998 [148], cuyo fin era precisamente la detección de intrusiones.

El NSL-KDD surge como una mejora para evitar los problemas del conjunto de datos KDD´99. Una de las mejoras sustanciales que tiene este nuevo conjunto de datos es la eliminación del gran número de registros redundantes. El incremento era de alrededor del 78% y 75% para los subconjuntos de entrenamiento y prueba respectivamente. Esta gran cantidad de registros redundantes en el conjunto de entrenamiento causa un aprendizaje inapropiado ya que se realiza un desvío hacia los registros más frecuentes, mientras que los registros menos infrecuentes como los *User to Root* (U2R) sean poco aprendidos y por consiguiente su probabilidad de detección sea casi nula por el método. KDD'99 [149] contiene alrededor de 4 GB de datos comprimidos de captura de tcpdump[4] en el programa de

[4] tcpdump. http://tcpdump.org

evaluación DARPA'98 IDS [150], lo que corresponde a alrededor de 7 semanas de tráfico de la red.

Las mejoras que presenta el NSL-KDD respecto a sus predecesores y que motivaron su selección para hacer este trabajo de investigación son:

- No incluye registros redundantes en la colección de datos para el entrenamiento, por lo tanto, los clasificadores no se sesgarán hacia los registros más frecuentes;

- No existen registros duplicados en las colecciones de datos propuestas para el test, haciendo que el rendimiento del aprendizaje no este sesgado por los métodos que tienen mejores tasas de detección de registros frecuentes;

- El número de registros seleccionados de cada grupo de nivel de dificultad es inversamente proporcional al porcentaje de registros en el conjunto original de datos KDD, dando como resultado que las tasas de clasificación de los distintos métodos de aprendizaje varíen en un rango más amplio, lo que hace que sea más eficiente para tener una evaluación precisa de las diferentes técnicas de aprendizaje;

- EL número de registros en las colecciones de datos para el entrenamiento y de test es razonable haciendo posible realizar experimentos con el conjunto de datos completo sin necesidad de seleccionar al azar una pequeña porción de éste.

Los tipos de ataques que contiene el *dataset* NSL-KDD se clasifican en cuatro categorías:

Denegación de servicio (*Denial of Service Attac*k, *DoS*). En este caso, el atacante intenta sobrecargar la máquina de la víctima para que esté demasiado ocupada para responder a nuevas peticiones legitimas. Así, la maquina atacada negará estas solicitudes. Este ataque se puede hacer por agotamiento de la memoria, la interfaz de red u otros recursos computacionales del servidor destino. Las clasificaciones de los ataques de DoS pueden apreciarse en la Tabla 6.1.

Tabla 6.1 Clasificación de ataques de Denegación de Servicios

Ataque	Descripción
Back	Ataque contra el servidor web Apache cuando un cliente pide una URL que contiene muchas barras.
Land	Envío de TCP/SYN falso con la dirección de la víctima como origen y destino, causando que se responda a sí mismo continuamente.
Neptune	Inundación por envíos de TCP/SYN en uno o más puertos.
Pod	Ping de la muerte, envía muchos paquetes ICMP muy pesados.
Smurf	El atacante envía un ping, que parece proceder de la víctima, en broadcast a una tercera parte de la red, donde todos los hosts responderán a la víctima.
Teardrop	Usa el algoritmo de fragmentación de paquetes IP para enviar paquetes corruptos a la víctima.

Usuario a Root (*User to Root attack*, *U2R*). Consiste en hacer una autenticación en el sistema como usuario normal, y luego tratar de cambiar a una credencial de superusuario mediante la explotación de vulnerabilidades presentes en el sistema. Después de hacer la explotación de la o las vulnerabilidades, el atacante obtiene acceso de administrador en el sistema. Las clasificaciones de los ataques de U2R pueden apreciarse en la Tabla 6.2.

Tabla 6.2 Clasificación de Ataques categoría de Usuario a Root

Ataque	Descripción
buffer_overflow	Desbordamiento de la pila del búfer.
Loadmodule	Ataque furtivo que reinicia la IFS para un usuario normal y crea un shell de root.
Perl	Establece el id de usuario como root en un script de perl y crea un shell de root.
Rootkit	Escenario de varios días donde un usuario instala componentes de un rootkit.

Remoto a Local (*Remote to Local Attack*, *R2L*): En este caso, el atacante utiliza vulnerabilidades del sistema al que no tiene acceso para iniciar una sesión en como un usuario normal. Este ataque puede ser un paso previo del ataque U2R. Las clasificaciones de los ataques de R2L se dan en la Tabla 6.3.

Tabla 6.3 Clasificación de Ataques categoría Remoto a Local

Ataque	Descripción
ftp_write	Usuario FTP remoto crea un archivo .rhost y obtiene un login local.
guess_passwd	Trata de adivinar la contraseña con telnet para la cuenta de visitante.
Imap	Desbordamiento remoto del búfer utilizando el puerto imap.
Multihop	Escenario de varios días donde el atacante primero accede a una máquina que luego usa como trampolín para atacar a otras máquinas.
Phf	Script CGI que permite ejecutar comandos en una máquina con un servidor web mal configurado.
Spy	Analizador de protocolos LAN por la interfaz de red.
Warezclient	Los usuarios descargan software ilegal publicado a través de FTP anónimo por el warezmaster.
Warezmaster	Subida FTP anónima de Warez (copias ilegales de software).

Ataque de Sondeo (*Probbing Attack*, *PROBE*). Este ataque se basa en tratar de recuperar información acerca de los ordenadores conectados a una red. El escaneo de puertos con el fin de tratar de descubrir puertos abiertos en los equipos que pertenecen a una red, es un ejemplo claro de este ataque. Las clasificaciones de los ataques de PROBE pueden apreciarse en la Tabla 6.4.

Tabla 6.4 Clasificación de Ataques categoría Probing

Ataque	Descripción
ipsweep	Sondeo con barrido de puertos o enviando pings a múltiples direcciones host.
nmap	Escaneo de redes mediante la herramienta nmap.
portsweep	Barrido de puertos para determinar qué servicios se apoyan en un único host.
satan	Herramienta de sondeo de redes que busca debilidades conocidas.

El conjunto de datos *KDD* en su nueva versión, el *dataset NSL-KDD* esta dispone para los investigadores en [151], y contiene características que se dividen en tres clases: características básicas, características de contenido, y características de tráfico. La principal razón para tener estos tres grupos de características es que la detección y la identificación de algunos ataques requiere el uso de más de una clase de características. Por ejemplo, las funciones basadas en el tiempo son necesarias para detectar los ataques que requieren algunas estadísticas calculadas durante un cierto periodo de tiempo. Por lo tanto, el primer paso que se realiza para poder trabajar con el conjunto de datos consiste en analizar el conjunto de datos con el fin de

extraer las características de los archivos de texto y construir los vectores que comprenden el espacio de características. A continuación, se dan algunos detalles de estos tres tipos de características:

1) **Características básicas**. En este primer grupo encontramos todas las categorías que se pueden extraer de una conexión TCP / IP.

2) **Características de tráfico**. Está compuesta por características calculadas con relación a un intervalo de la ventana y se divide en dos grupos. El primero de ellos es el grupo de los atributos de ***mismo host***, que tienen en cuenta solo las conexiones en los dos últimos segundos que tengan el mismo destino que la actual, y las estadísticas relacionadas con el protocolo, los servicios, etc. El segundo grupo es el de los atributos de ***mismo servicio***, que examinan sólo las conexiones en los dos últimos segundos que tienen el mismo servicio que la conexión actual.

   Este tipo de tráfico mencionado en las dos características anteriores se le conoce como características de tráfico basadas en tiempo. En la actualidad existen otro tipo de ataques llamados de ***sondeo*** que escanean los puertos de los hosts en un intervalo de tiempo mucho mayor que 2 segundos. Este tiempo mayor podría ser de 1 minuto, lo que da como resultado que no hay patrones de intrusión en una ventana de tiempo de 2 segundos. Para resolver este inconveniente, las características del ***mismo host*** y del ***mismo servicio*** se recalculan en una ventana de observación de 100 conexiones dando como resultado las características de tráfico.

3) **Características de contenido**. En este tipo de características encontramos a los ataques *R2L* y *U2R*. Estos ataques son sustancialmente diferentes que los ataques *DoS* y de sondeo ya que no cuentan con una frecuencia notoria de patrones secuenciales. Esto se debe a la cantidad de conexiones que hacen los ataques *DoS* y *Probing* a algún host en un periodo muy corto de tiempo. Sin embargo, los ataques *R2L* y *U2R* se encuentran incrustados en las porciones de datos de los paquetes, implicando en la mayoría de las veces una sola conexión. Para poder hacer una detección exitosa de este tipo de ataques se recurre a algunas características que hacen posible detectar algún comportamiento sospechoso en la parte de los datos. Un ejemplo claro es el número de intentos de acceso fallidos. A partir de aquí se generan las llamadas características de contenido.

En las Tablas 6.5, 6.6 y 6.7 se aprecia la distribución del conjunto de datos

NSL-KDD. Este contiene un total de 40 ataques los cuales se clasifican en 4 tipos (DoS, U2R, R2L y PROBE). Además, están las conexiones normales. En esta investigación se aplicó clasificación binaria, es decir a nivel de solo 2 categorías (ataque o normal), donde detecta si una conexión existente en el conjunto de datos es de tipo ataque, independiente del tipo de ataque que sea, o es de tipo normal, situación donde no existe amenaza alguna. Se crearon además ficheros equilibrados para los experimentos, lo que llevo a la utilización de conjuntos de datos de igual cantidad de ataques y conexiones normales en el fichero de entrenamiento, para no favorecer el sobre-entrenamiento por parte del modelo al favorecer a una categoría con más presencia de registros. El conjunto de entrenamiento contiene un 70% de conexiones, y el de test el 30% restante.

A continuación, se detallan las respectivas características con su descripción:

Tabla 6.5. Características básicas de las conexiones individuales TCP

Número de la característica	**Característica de datos de red**	**Descripción**
1	duration	length (number of seconds) of the connection
2	protocol_type	type of the protocol, e.g. tcp, udp, etc.
3	service	network service on the destination, e.g., http, telnet, etc.
4	src_bytes	number of data bytes from source to destination
5	dst_bytes	number of data bytes from destination to source
6	flag	normal or error status of the connection
7	land	1 if connection is from/to the same host/port; 0 otherwise
8	wrong_fragment	number of "wrong" fragments
9	urgent	number of urgent packets

Tabla 6.6. Características de trafico

Número de la característica	Característica de datos de red	Descripción
23	Count	number of connections to the same host as the current connection in the past two seconds
24	Sev_count	number of connections to the same service as the current connection in the past two seconds
25	serror_rate	% of connections that have "SYN'" errors
26	Sev_serror_rate	% of connections that have "SYN" errors
27	rerror_rate	% of connections that have "REJ" errors
28	Srv_error_rate	% of connections that have "REJ" errors
29	same_srv_rate	% of connections to the same service
30	diff_srv_rate	% of connections to different services
31	srv_diff_host_rate	% of connections to different hosts
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_server_rate	
39	dst_host_srv_serror_rate	
40	dst_host_rerror_rate	
41	dst_host_srv_rerror_rate	

Tabla 6.7. Características de contenido

Número de la característica	Característica de datos de red	Descripción
10	hot	number of "hot" indicators
11	num_failed_logins	number of failed login attempts
12	logged_in	1 if successfully logged in; 0 otherwise
13	num_compromised	number of "compromised" conditions
14	root_shell	1 if root shell is obtained; 0 otherwise
15	su_attempted	1 if "su root" command attempted; 0 otherwise
16	num_root	number of "root" accesses
17	num_file_creations	number of file creation operations
18	num_shells	number of shell prompts
19	num_access_files	number of operations on access control files
20	num_outbound_cmds	number of outbound commands in an ftp session
21	is_hot_login	1 if the login belongs to the "hot" list; 0 otherwise
22	is_guest_login	1 if the login is a "guest" login; 0 otherwise

Tabla 6.8. Ataques/Categorías/No. De registros del dataset NSL-KDD [152].

Ataque	Categoría	Frecuencia	Ataque	Categoría	Frecuencia
Apache2	DoS	737	Multihop	R2l	20
Back	DoS	555	Named	R2l	17
Land	DoS	8	Phf	R2l	4
Mailbomb	DoS	293	Sendmail	R2l	14
Neptune	DoS	12939	Snmpgetattack	R2l	178
Pod	DoS	79	Snmpguess	R2l	331
Processtable	DoS	685	Spy	R2l	1
Smurf	DoS	1194	Warezclient	R2l	181
Teardrop	DoS	200	Warezmaster	R2l	951
Udpstorm	DoS	2	Worm	R2l	2
Normal	Normal	23160	Xlock	R2l	9
Ipsweep	Probe	851	Xsnoop	R2l	4
Mscan	Probe	996	Buffer_Overflow	U2R	26
Nmap	Probe	374	Httptunnel	U2R	133
Portsweep	Probe	685	Loadmodule	U2R	3
Saint	Probe	319	Perl	U2R	2
Satan	Probe	1426	Ps	U2R	15
Ftp_write	R2l	4	Rootkit	U2R	17
Guess_Passwd	R2l	1241	Sqlattack	U2R	2
Imap	R2l	6	Xterm	U2R	13

Para realizar las comparaciones que permitan una adecuada interpretación del comportamiento del procedimiento propuesto, se realizaron experimentos con cinco categorías, siguiendo el método más usado en la literatura. Para ello se crean dos ficheros, el de test, del 30% de los datos, y el de entrenamiento del 70% del conjunto inicial de datos, usados para *testear* la clasificación y para *entrenar* el clasificador, respectivamente. En este caso puntual, el equilibrado no se hace efectivo para simular una situación más real.

Para tener una visión más aguda de la capacidad de detección de los clasificadores evaluados en este trabajo, se hicieron pruebas donde se crearon ficheros equilibrados para detectar, no solo la categoría del ataque, sino también un ataque concreto.

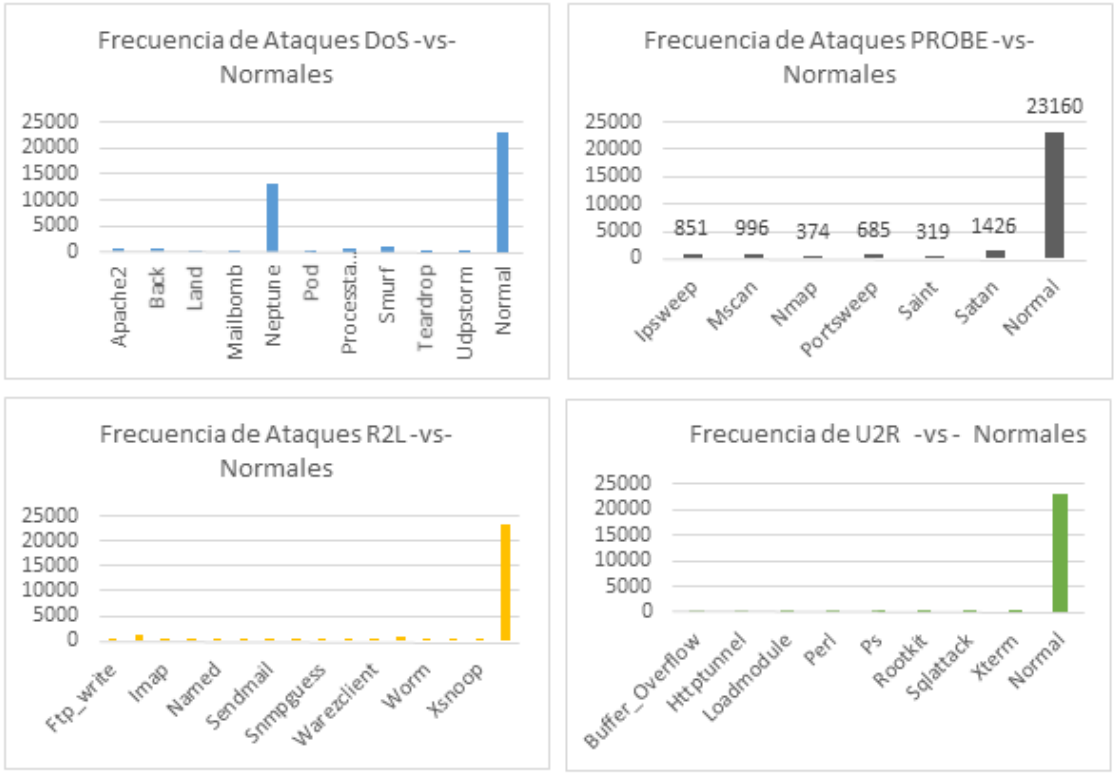


Figura 6.1. Frecuencias de ocurrencias de patrones anómalos y normales por categoría.

6.2 Pre-procesado de datos

Como se mencionó en el capítulo 2, en el diseño de IDS hay dos enfoques dependiendo de la filosofía de detección [153 - 156]. El primero es el llamado IDS basados en firmas [155], que analiza todos los paquetes entrantes en busca de patrones conocidos asociados con los intentos de intrusión. El segundo busca desviaciones de los patrones normales y decide si una conexión se clasifica como anómala, es decir basada en anomalías [156].

Estos sistemas suelen caracterizar los patrones normales y anomalías mediante técnicas de aprendizaje estadístico aplicada al tráfico de red. Algunas de ellas se han descrito en el capítulo 3 de esta tesis. Además, el uso de características complejas permite descubrir no sólo una intrusión real, sino también una potencial, como es el caso de los sistemas basados en anomalías que sirven de apoyo para identificar patrones normales a partir de patrones anormales. Sin embargo, debido a la diversificación de los ataques hoy día, es necesario utilizar características más complejas para mejorar la detección.

Como se ha dicho anteriormente, en el proceso de investigación realizado para este trabajo se implementa el PCA para generar un nuevo conjunto de características no correlacionadas con el fin de eliminar el ruido, y evitar el uso de variables de baja varianza. Por otra parte, estas nuevas funciones se seleccionan en función de su capacidad discriminativa. Acto seguido se modela el espacio de características, y se propone un método para dotar al SOM de un comportamiento probabilístico. Se trata así de obtener una versión difusa del SOM clásico expuesto en el apartado 4.5 de este trabajo, que permite medir la probabilidad de activación de cada unidad. Sin embargo, detectar un ataque, no es una tarea sencilla, y los enfoques anteriores independientemente de identificar su tipo, para obtener una buena precisión en relación a la detección por los valores de ataque [157, 158].

El pre-procesamiento de los datos de este trabajo de investigación comprende la codificación de las variables no continuas y la normalización. Esta etapa de pre-procesamiento es de vital importancia para la optimización del rendimiento del clasificador, sin embargo, pocos trabajos prestan suficiente atención a ella [159]. En la Sección 6.1 se describieron detalladamente los conjuntos de datos basados en KDD´99 y en NSL-KDD. Estos constan de 41 características, que se consideran suficientes para caracterizar conexiones anómalas. Además, existen tres grupos de características: continuas, simbólicas y binarias. Uno de los problemas encontrados en la etapa de pre-procesamiento es la clasificación de valores no numéricos, ya que la gran mayoría de los clasificadores solo toman valores numéricos obligando así a trasladar a un plano numérico todas aquellas características simbólicas. En la literatura [132, 159, 160] se hace una codificación básica simplemente sustituyendo cada característica simbólica por un entero. Aunque esto puede ser aceptable en muchas situaciones no es la mejor opción para codificar clasificadores basados en la distancia euclídea [161, 162]. En este trabajo se adopta una solución diferente, asignando cada característica simbólica a un sub-espacio de $\mathbb{R}^d$, donde d es el número de valores posibles de la variable discreta. Esta solución aumenta la dimensionalidad de los datos considerablemente. Un ejemplo es el caso de la característica "*service*" la cual tendría hasta 65 valores diferentes. No obstante, esto no es crítico para la propuesta del método de clasificación usado en este trabajo dado que, además, se usan técnicas de reducción de dimensionalidad para comprimir la información relevante con menos características. Esta técnica se utiliza con "*protocol*" (tres valores diferentes), "*service*" (65 valores diferentes), y "*port*" (cuatro características diferentes). Con la misma filosofía, las características binarias se dejan tal cual en el espacio de características.

6.2.1 Normalización del conjunto de datos

La normalización de datos permite que todas las características estén en una misma escala, de tal manera que no exista una característica que contribuya más que otra en la medida de distancia. Cuando se realiza el proceso de normalización se pueden presentar varios casos [153, 71]. De hecho, se presentan seis posibles implementaciones de métodos de normalización en la toolbox SOM de Matlab® [143]. Estas son *range*, *log*, *logistic*, *histD*, *histC* y *var*.

El método de normalización *range* es usado para escalar los valores de la variable entre $[0,1]$ mediante una transformación lineal simple. Los parámetros de esta transformación son los valores mínimos y el rango $\max(x) - \min(x)$ de la variable. Ahora bien, si la transformación es aplicada a nuevos datos con valores fuera del rango mínimo o máximo, los valores de la transformación estarán también fuera del rango $[0,1]$. La fórmula que muestra este método es:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{6.1}$$

El segundo método de normalización, *log*, se presenta como una transformación logarítmica importante cuando los valores de la variable se distribuyen de forma exponencial con valores demasiado pequeños y un número menor de valores grandes. Con esta trasformación lo que se busca es tener una mayor resolución en la parte baja del vector de componentes. Para ello se hace una transformación no lineal, donde ln es el logaritmo natural, generando valores no-negativos. Se debe tener cuidado al aplicar esta transformación a nuevos conjuntos de datos con valores por debajo de $\min(x) - 1$, ya que el resultado estará compuesto de números complejos. Así, lo más aconsejable es asignarle el valor mínimo $\min(x)$ de manera manual en el caso de que se conozcan los valores previamente. La ecuación que se aplica en este método es:

$$x' = ln\,(x - \min(x) + 1) \tag{6.2}$$

La implementación *logistic* (también es conocida como *softmax*), asegura que todos los valores, desde menos infinito hasta más infinito se encuentren dentro del rango $[0,1]$. Esta transformación es lineal alrededor del valor medio, y aplica una no linealidad suavizada en ambos extremos que asegura que todos los valores estén dentro del rango. El primer dato se escala

usando la fórmula de la normalización tipo *var* y luego se aplica la función *logistic* así:

$$x' = \frac{1}{(1 + e^{-\hat{x}})} \tag{6.3}$$

donde x es el valor medio y r la desviación estándar de los valores tal como en el método de normalización *var*.

Para el caso de *histD* o equalización de histograma discreto, se ordenan los valores y después se cambia cada uno por su número ordinal. Finalmente se hace un escalado de los valores de modo que estén en el rango de $[0,1]$. Este método es útil cuando se usan variables tanto continuas como discretas. Sin embargo, como los parámetros de transformación son todos valores únicos del conjunto de datos de inicialización, podría considerarse el uso de una alta cantidad de memoria y coste computacional. Ahora bien, si la variable tomara pocos valores es recomendable el uso del método *histC*. El método *histD* si es aplicado a valores que no sean parte del conjunto de valores originales no es exactamente reversible.

A diferencia de *histD*, *histC* es una equalización de histograma contínua que utiliza una transformación lineal parcial. El rango de valores se divide en una serie de contenedores de tal forma que el número de valores en cada contenedor es "casi" el mismo. Los valores son transformados linealmente en cada contenedor y finalmente todos los valores son linealmente escalados entre $[0,1]$. El número de contenedores será el redondeo de la raíz cuadrada del número de valores únicos en el conjunto de inicialización. Aunque la equalización del histograma resultante no es tan buena como la hecha mediante *histD*, este método tiene a su favor el poder ser reversible de manera exacta, incluso fuera del rango de valores originales.

Para esta tesis se implementó el método de normalización *var*. Debido a la naturaleza de las características del conjunto de dados utilizado, que son tanto numéricas como categóricas, el proceso de normalización se aborda de manera diferente en cada caso. Las variables continuas se normalizan a media cero y varianza única. Se trata de una transformación lineal simple como se observa en la siguiente ecuación:

$$\hat{x} = \frac{x - \bar{x}}{\sigma} \tag{6.4}$$

donde x y σ son la media y la desviación estándar de la variable x,

respectivamente. Esto equivale a la expresión de la variable x como el número de desviaciones estándar de distancia de la media. Sin embargo, las variables categóricas requieren un tratamiento diferente. Específicamente, estas variables deben ser codificadas antes de la normalización de acuerdo con la medida de similitud [161]. Algunos trabajos utilizan una codificación simple de ***categoría-a-numero***, pero esta falla cuando se calcula la medida de similitud [153, 159]. Una evidencia de lo anterior se presenta con la característica "*protocol*" que no puede ser codificada usando números consecutivos, ya que indicará una similitud entre diferentes protocolos (por ejemplo, TCP y UDP). Debido a este problema, se usó la siguiente medida de similitud $s(x_k, y_k) = 1$ si $x_k = y_k$ y $s(x_k, y_k) = 0$ en caso contrario.

Esto que la codificación de estas características se hace como si fueran vectores binarios, en el que cada componente indica la activación de la característica correspondiente (por ejemplo, si el protocolo es tcp o no en una conexión especifica). Este proceso tiene repercusiones muy importantes, ya que se logró pasar de 41 características a 38. Estas 3 características que se restan son "service", "flag" y "protocol", teniendo "service" la posibilidad de usar 65 posibles valores, "flag" hasta 11 valores y "protocol" un total de 3 valores. Cada uno de los valores de las características simbólicas se toman como características binarias que no tienen que normalizarse. Estas características no son normalizadas.

6.3 Métodos de Validación cruzada

Existen diferentes posibilidades que son aplicables para mejorar el rendimiento de los clasificadores descritos en la sección 3.4. Éstas, son propuestas que utilizan la reordenación de los datos y antes de acceder a la entrada, hacen que la clasificación sea más eficiente a la hora de evaluar las muestras por los distintos clasificadores, por haber tenido un mejor entrenamiento. La estratificación y la validación cruzada son aplicaciones que sirven para esta índole.

La *estratificación* es el proceso mediante el cual se hace un proceso de heterogenización del conjunto de muestras de clasificación y entrenamiento. La estratificación hace una división inicial del conjunto de datos en subconjuntos disjuntos con una distribución heterogénea de clases. En la mayoría de los casos este proceso se hace debido al gran tamaño de los conjuntos de muestra, particionandolos generalmente a 2/3 para el entrenamiento y 1/3 para la etapa de clasificación. Esto quiere decir que, si tuviésemos 900 muestras ordenadas por la clase a la que pertenecen, con 10

clases debidamente identificadas, se entrenarían las 600 primeras y al estar ordenadas por clases, es posible que alguna clase no sea entrenada en este proceso. Lo anterior dará lugar a que en la fase de clasificación se encuentren algunas de las clases que no se han entrenado previamente y viceversa, aumentando significativamente el porcentaje de error en la decisión final.

Estos subconjuntos son independientes de los demás, y la cantidad de muestras que tenga determinará su tamaño. Al utilizar un número adecuado de subconjuntos se podrá reducir de forma significativa el tiempo de ejecución, lo que permitirá ejecutar más recursos por parte del algoritmo que se vaya a usar [163] mejorando el rendimiento y por ende el porcentaje de acierto.

Teniendo en cuenta lo anterior, en la estratificación, el conjunto de datos inicial C es dividido en N subconjuntos disjuntos de igual tamaño $C = \{X_1, X_2, \dots, X_N\}$. La distribución de las clases en el proceso de partición se mantiene, y el conjunto de datos que se entrena Y será complementario a $X, Y = C/X$. Por tanto, cuando un conjunto de datos ésta dividido en grupos homogéneos, usando estratificación, se producirá un intercambio de elementos entre estos grupos con el fin de hacer una distribución heterogénea. Para un grupo de muestras pequeño y con un número elevado de clases la estratificación es de uso indispensable. El papel de la estratificación es entonces un agrupamiento inteligente de los elementos que componen la población de determinado subconjunto.

Según Zseby [164] la utilización de subconjuntos de igual tamaño da resultados similares a utilizarlos de diferentes tamaños. Sin embargo, existe la manera de mejorar la precisión de este proceso si se hace una búsqueda de los límites entre los subconjuntos de forma automática utilizando un análisis de clúster, eligiendo limites óptimos entre estos en lugar de hacerlos de tamaño fijo o fijarlos manualmente [165].

A continuación, se muestra cómo se pasa de un conjunto de 900 muestras ordenadas por clase, diferenciadas 10 colores, al mismo conjunto de muestras después de aplicar la técnica de la estratificación.

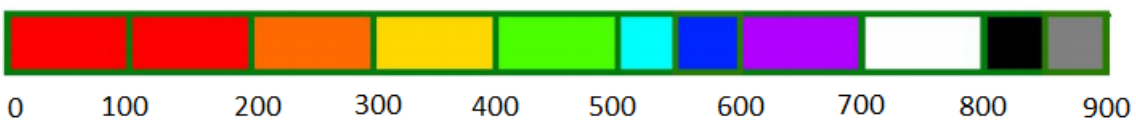


Figura 6.2. Muestras sin estratificación.

Figura 6.3. Muestras estratificadas.

En la figura 6.2 se ve el conjunto de 900 muestras bien diferenciadas por clases, donde las primeras 200 pertenecen a la clase1, de color rojo, las 100 siguientes representadas con color naranja representan la segunda clase y así sucesivamente. Al no usar estratificación, la etapa de entrenamiento tomará las primeras 600 muestras (como se dijo anteriormente 2/3 del total), lo que repercute en un entrenamiento nulo de cuatro de las diez clases existentes. Este hecho se repetirá al hacer la clasificación, ya que las 300 muestras a clasificar será de 1/3 restante, y el número de aciertos será nulo.

En la figura 6.3, se observa como la estratificación de las muestras divide de forma heterogénea los datos, para que al momento de entrenar como de clasificar todas las clases del conjunto de muestras se puedan evaluar a la vez.

La teoría de la *validación cruzada* fue originalmente desarrollada por Geisser [166]. En su trabajo muestra la importancia de vigilar la posible presencia del error estadístico tipo III [167] en cualquier proceso de decisión. Este error es el causante de rechazar la hipótesis nula de manera correcta por razones erróneas. Un ejemplo de esto se evidencia cuando el espacio muestral es limitado.

También llamada estimación por rotación según [168, 169], la validación cruzada es una técnica utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba. Consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones. Se utiliza en entornos donde el objetivo principal es la predicción y se quiere estimar cómo de preciso es un modelo [170].

El objetivo principal de la validación cruzada se basa en un proceso de ajuste a un modelo con uno o más parámetros de ajuste desconocidos, y unos datos de entrenamiento que se desea analizar. Este proceso de ajuste optimiza los parámetros del modelo con el fin de ajustar a los datos de entrenamiento tan bien como sea posible. Uno de los conceptos más comunes que ocurre cuando el tamaño de los datos de entrenamiento es pequeño, o cuando el número de parámetros del modelo es grande es el sobreajuste, que ocurre cuando se toma una muestra independiente como dato de prueba (validación) del mismo grupo que los datos de entrenamiento. Para este caso puntual, el

modelo no se ajustará a los datos de prueba igual de bien que a los datos de entrenamiento. La validación cruzada es una manera de predecir el ajuste de un modelo a un hipotético conjunto de datos de prueba cuando no disponemos del conjunto explícito de datos de prueba [171].

Para aplicarla, se divide el conjunto muestral en subconjuntos complementarios. Se realiza el análisis en uno de los subconjuntos de esta división llamado comúnmente conjunto de entrenamiento, para luego validar éste análisis en otro subconjunto denominado subconjunto de validación o *test*. La aplicación de la validación cruzada de manera reiterativa usando distintas particiones y promediando los resultados de dicha validación sobre todas las iteraciones reduce la variabilidad en la evaluación global de la generalización, siendo esta una de las fortalezas de este método. Cuando se aplica validación cruzada, el conjunto llamado *test* no evidencia un *test* "veraz" debido a que la etiqueta de los elementos del conjunto "*test*" es conocida. Aplicando lo anterior se puede comparar el resultado de *test* con la etiqueta original determinando así si es un VP, FP, VN o un FN.

A continuación, se describen algunos de los métodos de validación cruzada más destacados.

6.3.1 Validación por Sub-Muestreo aleatorio

Este método consiste en dividir aleatoriamente el conjunto de datos de entrenamiento y el conjunto de datos de prueba. Para cada división la función de aproximación se ajusta a partir de los datos de entrenamiento y calcula los valores de salida para el conjunto de datos de prueba. El resultado final se corresponde con la media aritmética de los valores obtenidos para las diferentes divisiones. La principal ventaja de este método es que la división de datos *entrenamiento-prueba* no depende del número de iteraciones. Pero, en cambio, puede existir un solapamiento de los subconjuntos de prueba y entrenamiento debido a que unas muestras pueden evaluarse más de una vez y otras que no son evaluadas debido a la aleatoriedad de la selección de las muestras.

Para el caso de la validación cruzada aleatoria, a diferencia del método de los *K*-Pliegues que se describe en el siguiente apartado, se toman muestras al azar durante *k* iteraciones, aunque de igual manera, se realiza un cálculo de error para cada iteración. El resultado final también se obtenemos a partir de realizar la media aritmética de los K valores de errores obtenidos, según la misma fórmula:

$$E = \frac{1}{K}\sum_{i=1}^{K} E_i \tag{6.5}$$

6.3.2 Validación dejar uno fuera

Este método es uno de los 4 explicados por Rudys en [172], llamado así por las siglas en ingles LOOCV "*Leave-one-out cross-validation*" se encarga de separar los datos de forma tal que para cada iteración tengamos una sola muestra para los datos de prueba y todo el resto conformando los datos de entrenamiento. El proceso se repite N de veces de manera que todas las muestras se usan una vez como observaciones para la validación, considerándose así un caso particular del método anterior de validación. La evaluación viene dada por el error, y en este tipo de validación cruzada el error es muy bajo, pero en cambio, a costo computacional es muy elevado, puesto que se tiene que realizar un elevado número de iteraciones, tantas como muestras tengamos y para cada una analizar los datos tanto de entrenamiento como de prueba.

El resultado final del error lo obtenemos realizando la media aritmética de los N valores de errores obtenidos, según la fórmula:

$$E = \frac{1}{N}\sum_{i=1}^{N} E_i \tag{6.6}$$

6.3.3 Validación Cruzada K-Pliegues

Breiman [173] propone este método de validación como $K - Pliegues$ o K iteraciones, donde el conjunto muestral original se divide en K subconjuntos. Puesto que los datos son a menudo escasos, esto no suele ser posible. Por ello, esta variante de la validación cruzada utiliza parte de los datos disponibles para ajustar el modelo, y una parte distinta para comprobarlo. Esta técnica es considerada una media fiable que ofrece buena complejidad en tiempo real [174]. El modelo, al repetir K veces el proceso de entrenamiento, produce un porcentaje de acierto resultante que es utilizado como una medida de bondad del algoritmo evaluado [175].

La definición matemática se denota así: si C es el conjunto de datos, cada partición aleatoria de muestras X contendrá el mismo número de datos que

las demás, y los subconjuntos resultantes son $X_1, X_2, \ldots, X_N$. El pseudocodigo de esta notación puede ser expresado así:

For $K = 1$ to N
    X_K se evalúan y entrenan $C - X_K$
    n_K = resultado para X_K
Devuelve $(n_1 + n_2 + \cdots + n_N)\ N$

Pseudocodigo 6.1

donde se evalúan todos los subconjuntos X_K desde el primero hasta el último, se clasifica el subconjunto X_K, entrenando el resto para incrementar el aprendizaje $(C - X_K)$. Este proceso se realizará las veces que sea necesario con el único objetivo de analizar todas las muestras, extrayéndose la media al calcular los resultados de la clasificación de cada uno de los subconjuntos.

En la figura 6.4 se muestra un ejemplo de la implementación de la validación cruzada con $K = 10$, realizando una división en 10 subconjuntos (N =10) para 1000 muestras. Ejecutando el pseudocodigo 6.1, se tendrían las siguientes instrucciones:

- Para cada iteración:
  - Se entrenan todos los bloques menos el de color verde, este será el bloque utilizado para hacer el proceso de clasificación.
  - Devuelve el resultado obtenido
- Al finalizar el proceso:
  - Se realiza una media de los resultados extraídos en cada una de las siguientes iteraciones para calcular el porcentaje de acierto del algoritmo.

Figura 6.4. Diez particiones que utilizan 10-fold Cross-Validation.

Este método es muy preciso puesto que evaluamos a partir de *K* combinaciones de datos de entrenamiento y de prueba, pero aun así tiene una desventaja, y es que es lento desde el punto de vista computacional. En la práctica, la elección del número de iteraciones depende de la medida del conjunto de datos. Lo más común es utilizar la validación cruzada de 10 iteraciones (*10-fold cross-validation*). Este método es de gran apoyo cuando encontramos un número elevado de elementos en la muestra, o cuando el costo de los algoritmos de clasificación es alto, de tal forma que se tiene control sobre el número de veces que se itera la validación a través del número *K*. Para este caso puntual se puede considerar que en cada pliegue se contenga la misma proporción de etiquetas o respuestas.

La evaluación de las diferentes validaciones cruzadas normalmente viene dada por el error obtenido en cada iteración, para el caso de *K*-pliegues se obtiene a partir de realizar la media aritmética de los K valores de errores

obtenidos, según la fórmula (6.5):

$$E = \frac{1}{K}\sum_{i=1}^{K} E_i$$

Es decir, se realiza el sumatorio de los K valores de error y se divide entre el valor de K.

6.3.4 Significancia estadística

En esta tesis, gracias a la existencia de muestras suficientes para mantener la proporción entre las clases existentes, la configuración experimental se ha hecho con validación cruzada, concretamente con $K - fold$ $(K = 10)$ usando el método de validación cruzada con estratificación (*estratified cross-validation*) asegurando que la proporción de las dos clases (conexiones normales y anómalas) se conserva en cada pliegue, siendo este un método popular para estimar el error de generalización. En efecto, este error siempre dará lugar a una sobreestimación de la predicción del error verdadero. Esta sobreestimación dependerá de la pendiente de la curva de aprendizaje del clasificador y reduciéndose cuando K aumenta.

El error estándar se calcula a partir de la desviación estándar. Validaciones cruzadas llevadas a cabo por $K << N$ permitirá estimar la desviación estándar de un experimento $CV(\zeta)$. Primero, el error de validación en él $j - th\ fold$ se promedia como

$$CV_j(\zeta) = \frac{1}{n_j} e_j(\zeta) = \frac{1}{n_j}\sum_{i \in F_j} \left(y_i - \hat{f}_{\zeta}^{-j}\,(x_i)\right)^2 \qquad (6.7)$$

donde n_j representa la varianza de la variable x. Finalmente, el error estándar (o desviación estándar de $CV_j(\zeta)$) se calcula como:

$$SEM(\zeta) = k^{-\frac{1}{2}} SD(\zeta) \qquad (6.8)$$

6.4 Conclusiones

En este capítulo se detalló el conjunto de datos a utilizar en los experimentos, así como las diferentes formas de normalización posibles, tomando como base para esta tesis la normalización *var*.

Hemos hecho un análisis detallado de los métodos de validación cruzada, ya que este concepto se aplica en la configuración experimental del conjunto de datos de esta tesis. El método escogido para ser implementado es la validación cruzada por *K-pliegues* siendo el valor de K=10.

En el siguiente capítulo se explica detalladamente los métodos propuestos de selección y clasificación de características para detección de anomalías que implementa este trabajo de investigación y se proporcionan los resultados experimentales que nos permiten evaluar las prestaciones del método propuesto.

Capítulo 7. Resultados Experimentales

En este capítulo se muestra con detalle los experimentos realizados. Por un lado, se determina el número de características que maximizan el rendimiento de clasificación y por otro se obtiene el número de unidades de SOM para maximizar el rendimiento. Igualmente se detallan una serie de pruebas realizadas con las que se logró recopilar información relevante para la toma de decisiones en la implementación definitiva de los experimentos como fue el tamaño del mapa a utilizar y el número de características utilizadas.

Como se ha dicho, el método propuesto se ha evaluado mediante el uso de los datos de entrenamiento y *test* del conjunto de datos NSL-KDD como subconjuntos separados. Por lo tanto, no es necesario extraer subconjuntos de bases de datos para realizar una validación cruzada. Consecuentemente, los datos del *test* no se utilizan en el proceso de *training* de ninguna manera. Por otra parte, la información de la etiqueta del conjunto *test* solo se utiliza para la evaluación del rendimiento. El rendimiento de la clasificación ha sido evaluado mediante el cálculo de tres medidas estadísticas: la sensibilidad, la especificidad y la precisión, que fueron explicadas en la sección 4.6, así como mediante el cálculo de las curvas ROC.

7.1 Pruebas preliminares

Para configurar los escenarios de experimentación, de cara a obtener los resultados de los mismos que dan sustento a esta tesis, se realizaron tres grupos de experimentaciones preliminares con el fin de analizar tendencias y comportamientos en el conjunto de datos. El primero de ellos implicó pruebas de detección de anomalías de red con el Factor Discriminante de Fisher (FDR) para la selección de características y Mapas Autoorganizativos (SOM). La cantidad de características que se usaron en el conjunto de datos se fue variando, empezando con las primeras 5 e incrementando de 5 en 5 hasta tener un total de 35 características, tratando de buscar la mejor aproximación al número mínimo de características que evidenciaran un buen desempeño del clasificador en el caso biclase.

El segundo escenario de pruebas permitió contrastar las diferentes métricas de calidad expuestas en la Sección 3.6 aplicando las técnicas de clasificación no supervisadas SOM y el modelado GMM. Se aplicó la misma variabilidad de características que se usó para el primer experimento con el mismo objetivo en mente.

Los resultados de estos experimentos se pueden apreciar en las tablas 7.1 y 7.2 para FDR-SOM, y 7.3 y 7.4 para FDR-SOM-GMM, respectivamente. Resultado de este conjunto de procedimientos, se logró identificar qué tamaños de los mapas eran los más relevantes con miras a experimentos posteriores, siendo los tamaños más representativos: 4x4, 6x6, 8x8 y 10,10. Las medidas estadísticas a valorar fueron: la sensibilidad, la especificidad y la precisión.

Tabla. 7.1. Mejores resultados obtenidos para diferentes métodos de clasificación

Tamaño	**Medida estadística**	**FDR-SOM**		
		5	**10**	**15**
10x10	Sensibilidad	30.17	45.69	43.70
	Especificidad	99.87	98.79	99.73
	Precisión	0.60	0.68	0.67
8x8	Sensibilidad	27.96	35.44	**45.40**
	Especificidad	99.88	98.81	**99.27**
	Precisión	0.58	0.62	**0.68**
6x6	Sensibilidad	31.32	34.72	33.11
	Especificidad	99.86	99.49	98.57
	Precisión	0.60	0.62	0.61
4x4	Sensibilidad	43.01	33.58	35.11
	Especificidad	98.93	99.62	98.59
	Precisión	0.67	0.62	0.62

Tabla. 7.2. Mejores resultados obtenidos para diferentes métodos de clasificación

Tamaño	Medida estadística	FDR-SOM		
		20	25	30
10x10	Sensibilidad	39.87	97.43	8.76
	Especificidad	98.63	90.18	89.86
	Precisión	0.65	0.55	0.43
8x8	Sensibilidad	31.90	96.39	9.79
	Especificidad	99.78	90.18	89.85
	Precisión	0.61	0.50	0.44
6x6	Sensibilidad	32.21	89.89	10.0
	Especificidad	99.56	90.71	89.37
	Precisión	0.61	0,51	0.44
4x4	Sensibilidad	33.08	92.11	10.0
	Especificidad	99.50	90.32	89.37
	Precisión	0.61	0.52	0.44

Tabla. 7.3. Mejores resultados obtenidos para diferentes métodos de clasificación

Tamaño	Medida estadística	FDR-SOM+GMM		
		5	10	15
10x10	Sensibilidad	23.01	47.60	46.87
	Especificidad	99.91	97.73	97.99
	Precisión	0.56	0.69	0.68
8x8	Sensibilidad	27.63	37.37	**50.52**
	Especificidad	99.87	97.89	**97.60**
	Precisión	0.58	0.63	**0.70**
6x6	Sensibilidad	31.00	33.94	32.17
	Especificidad	99.87	99.43	99.22
	Precisión	0.60	0.62	0.61
4x4	Sensibilidad	43.91	34.06	37.04
	Especificidad	98.78	99.45	97.80
	Precisión	0.67	0.62	0.63

Tabla. 7.4. Mejores resultados obtenidos para diferentes métodos de clasificación

Tamaño	Medida estadística	FDR-SOM+GMM		
		20	25	30
10x10	Sensibilidad	42.14	20.55	11.62
	Especificidad	99.14	91.17	60.53
	Precisión	0.66	0.42	0.37
8x8	Sensibilidad	46.29	25.13	23.04
	Especificidad	99.48	90.81	48.13
	Precisión	0.69	0.48	0.34
6x6	Sensibilidad	30.79	29.11	30.87
	Especificidad	99.47	88.87	64.14
	Precisión	0.69	0.66	0.43
4x4	Sensibilidad	35.32	30.72	29.76
	Especificidad	99.26	98.31	61.98
	Precisión	0.62	0.59	0.47

Resultado de este conjunto de procedimientos, se logró identificar dentro de las pruebas que de los tamaños del mapa utilizado (4x4, 6x6, 8x8 y 10x10), el más óptimo en estos experimentos preliminares fue el tamaño de 8x8. En la tabla 7.2 se observan las medidas estadísticas valoradas y los resultados que se obtuvieron para la variación de la cantidad de características. Al finalizar el experimento, la cantidad que representó el comportamiento más ideal fue el de 15 características. Estos datos sirvieron como base para los escenarios de experimentación de la tesis.

El tercer escenario que se evaluó, fue la representación en los respectivos SOM, de las probabilidades logarítmicas apriori de activación para cada unidad, basándose en la metodología de inicialización del SOM implementada en la Sección 5.3.2. En las figuras 7.1, 7.2 y 7.3 se muestran tres gráficas generadas al final de la fase de entrenamiento, usando clasificación multiclase con cinco categorías (Normal, DoS, Probe, U2R y R2L). Con la notación siguiente: Normales=círculos azules, DoS=cuadrados rojos, PROBE=cuadrados verdes, U2R=cuadrados negros, R2L=cuadrados naranjas.

Para el escenario de la Figura 7.1 se puede ver que las muestras normales activan la mayoría de las unidades normales (círculos azules), las muestras de patrones de la clase DoS (cuadrados rojos) y PROBE (cuadrados verdes) son relativamente bajos, mientras que los patrones de U2R (cuadrados negros) y R2L (cuadrados naranjas) son nulos.

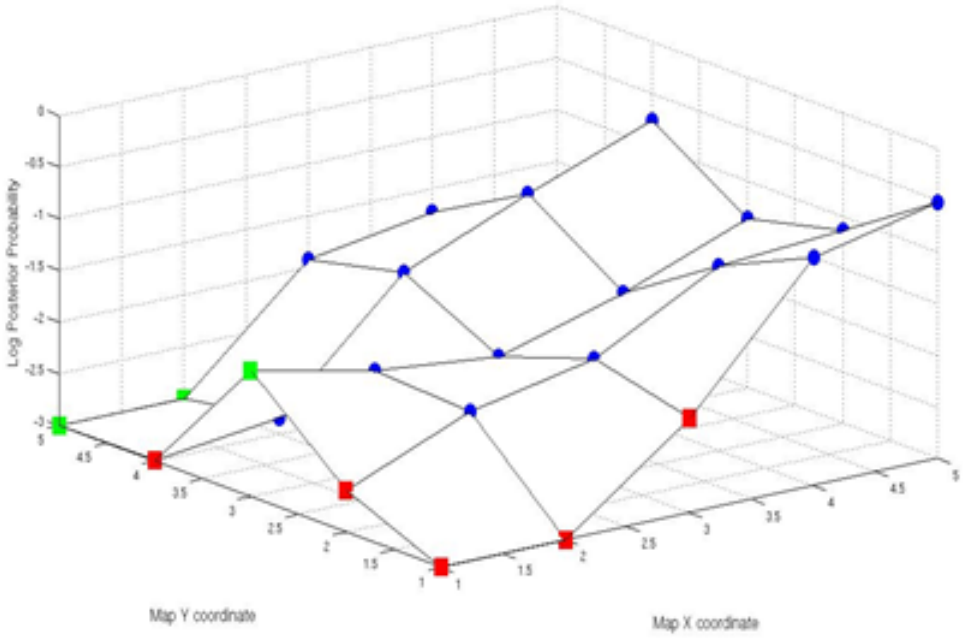


Figura. 7.1. SOM de activación de las probabilidades logarítmicas a priori para cada unidad EXPERIMENTO 1.

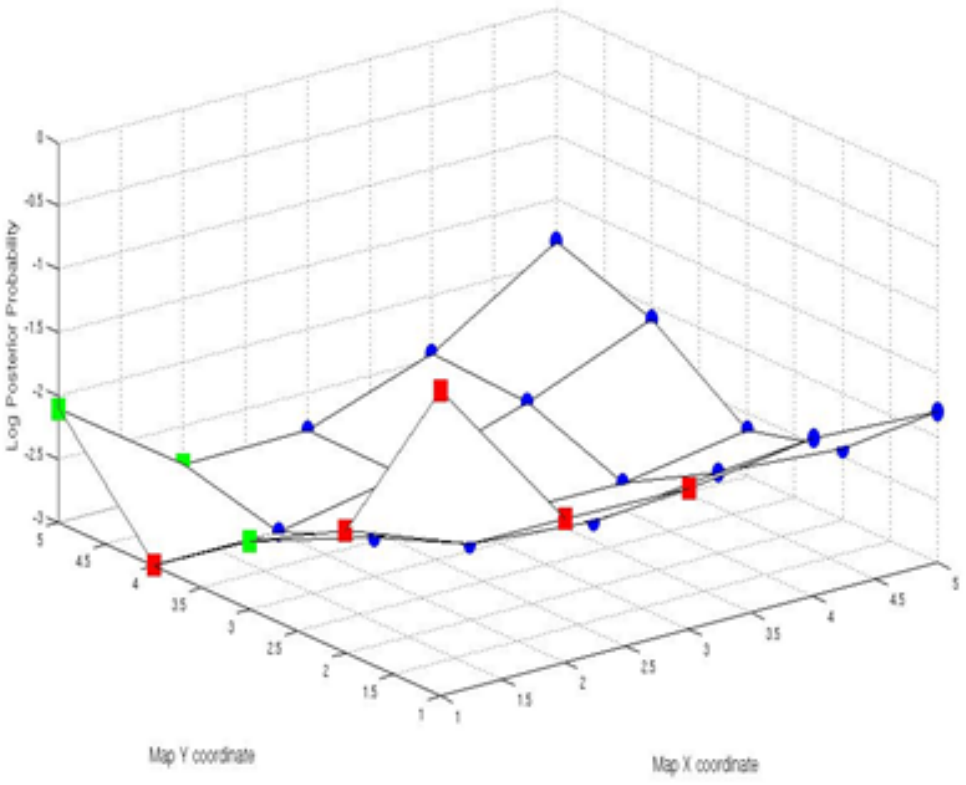


Figura. 7.2. SOM de activación de las probabilidades logarítmicas a priori para cada unidad EXPERIMENTO 2.

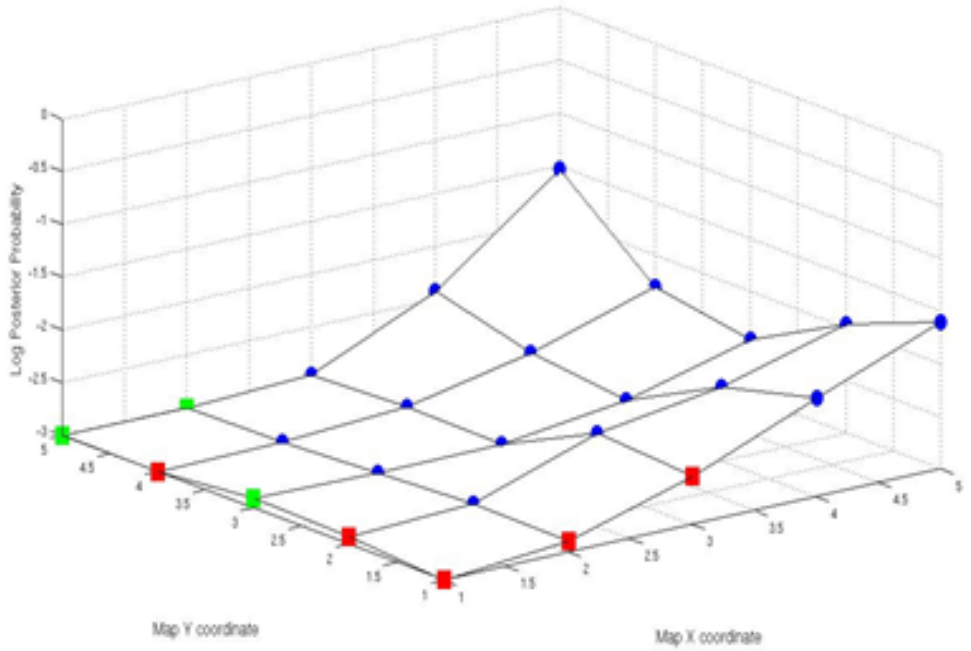


Figura. 7.3. SOM de activación de las probabilidades logarítmicas a priori para cada unidad EXPERIMENTO 3.

Para el escenario de la Figura 7.2 se puede comprobar que las muestras anómalas activaron las unidades que efectivamente representan anomalías en una porción mayor a la Figura 7.1. Para este caso puntual, las muestras de patrones de la clase DoS (cuadrados rojos) y PROBE (cuadrados verdes) crecieron, los patrones de la clase Normal (círculos azules) bajaron su intensidad de ocurrencia, mientras que los patrones de U2R (cuadrados negros) y R2L (cuadrados naranjas) siguieron siendo nulos.

Para el escenario de la Figura 7.3 se puede ver que las muestras normales activan la mayoría de las unidades normales (círculos azules), manteniendo gran parte de las unidades que representan anomalías desactivadas. Los patrones de U2R (cuadrados negros) y R2L (cuadrados naranjas) con nulos.

Aunque la clasificación multiclase usa cinco tipos de patrones (4 anómalos y 1 normales), representados como Normales=Círculos Azules, DoS=Cuadrados Rojos, Probe=Cuadrados Verdes, U2R=Cuadrados Negros, R2L=Cuadrados Naranja, en las gráficas resultantes de los experimentos se logra apreciar la ausencia de algunos de estos patrones, como son U2R=Cuadrados Negros y R2L=Cuadrados Naranja. Este hecho es el resultado de una presencia insuficiente de este tipo de patrones tanto en el conjunto de entrenamiento como en el de prueba, lo que ocasiona a un

entrenamiento no tan eficiente del clasificador. En nuestro caso, la distribución del conjunto de datos NSL-KDD se encuentra dada en la Figura 7.4.

Archivos del NSL-KDD
KDDTrain+.arff
KDDTrain+.txt
KDDTrain+_20Percent.arff
KDDTrain+_20Percent.txt
KDDTest+.arff
KDDTest+.txt
KDDTest-21.arff
KDDTest-21.txt

Training			Test		
Clase	Patrones	%	Clase	Patrones	%
Normal	67.343	53.47	Normal	9.711	43.08
DoS	45.927	36.46	DoS	7.458	33.08
Probe	11.656	9.25	Probe	2.421	10.74
R2L	995	0.79	R2L	2.754	12.22
U2R	52	0.04	U2R	200	0.89
Total	**125.937**	**100,00**	**Total**	**22.544**	**100,00**
R2L (Spy y Warezclient)			**DoS** (apache2, mailbomb, processtable y udpstorm) - **PROBE** (mscan y saint)- **R2L** (httptunnel, named, sedmail, snmpgetattack, xlock y xsnoop) - **U2R** (ps, snmpguess, sqlattack, worm y xterm)		

Figura. 7.4. SOM de activación de las probabilidades logarítmicas a priori para cada unidad (MULTICLASE c).

8.2 Escenarios de experimentación

Con base en las pruebas preliminares realizadas, se han elaborado dos experimentos diferentes. El primero tiene como objetivo determinar el número de características que maximizan el rendimiento de clasificación, con miras a darle al clasificador un poco más de relevancia al momento de

la clasificación. Para ello, en esta primera fase, se hicieron dos grupos de experimentos con los métodos de selección y clasificación de características expuestos en el capítulo 6. El primero de ellos, para los casos de PCA, FDR y para la unión de estos dos métodos (PCA+FDR), se evaluó el número de características que proporcionan el mejor rendimiento en términos de sensibilidad, especificidad y exactitud.

Los resultados que arrojo PCA del procedimiento anteriormente mencionado puede verse en la Figura 7.5.

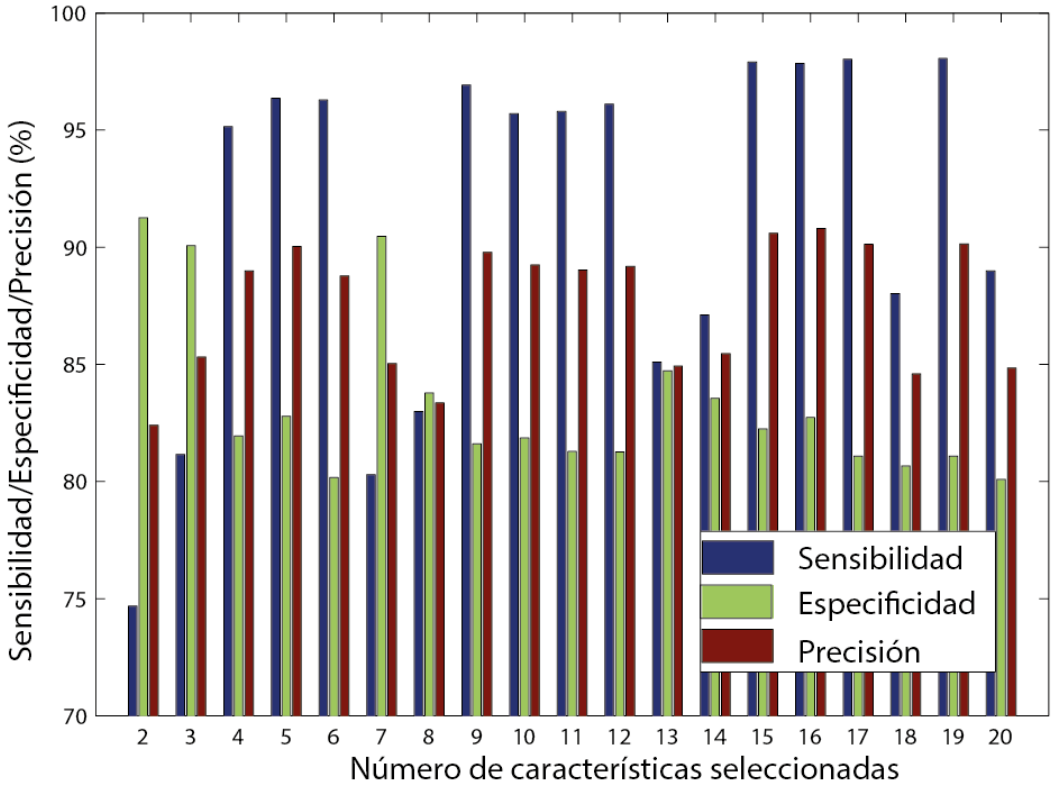


Figura.7.5. Resultados de la clasificación usando PCA en función del número de características seleccionadas.

Para el caso de FDR, los resultados pueden verse en la Figura 7.6.

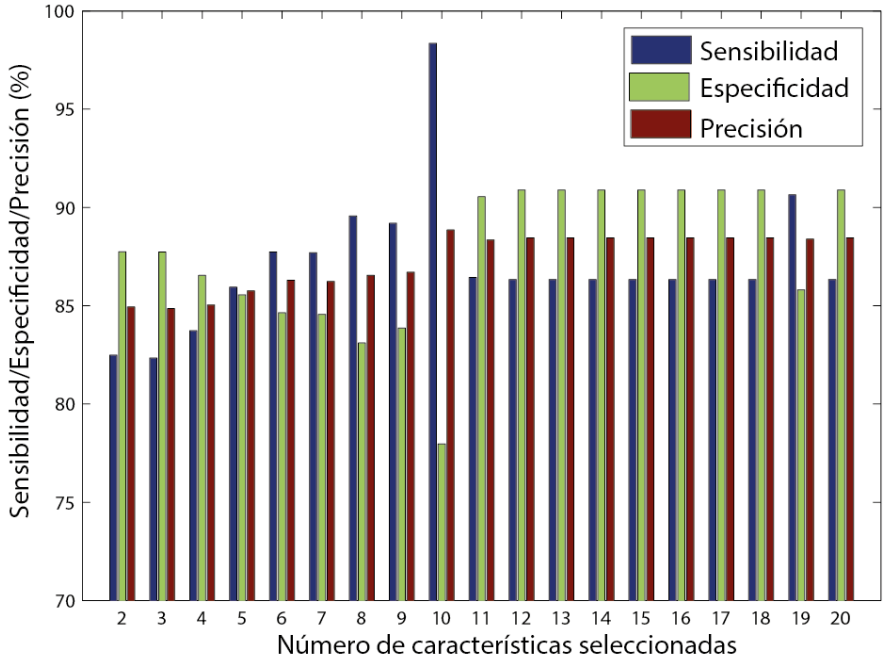


Figura.7.6. Resultados de la clasificación usando FDR en función del número de características seleccionadas.

La implementación de hacer en una primera instancia un filtrado con el método PCA y luego una selección de características usando FDR, se muestra en Figura 7.7.

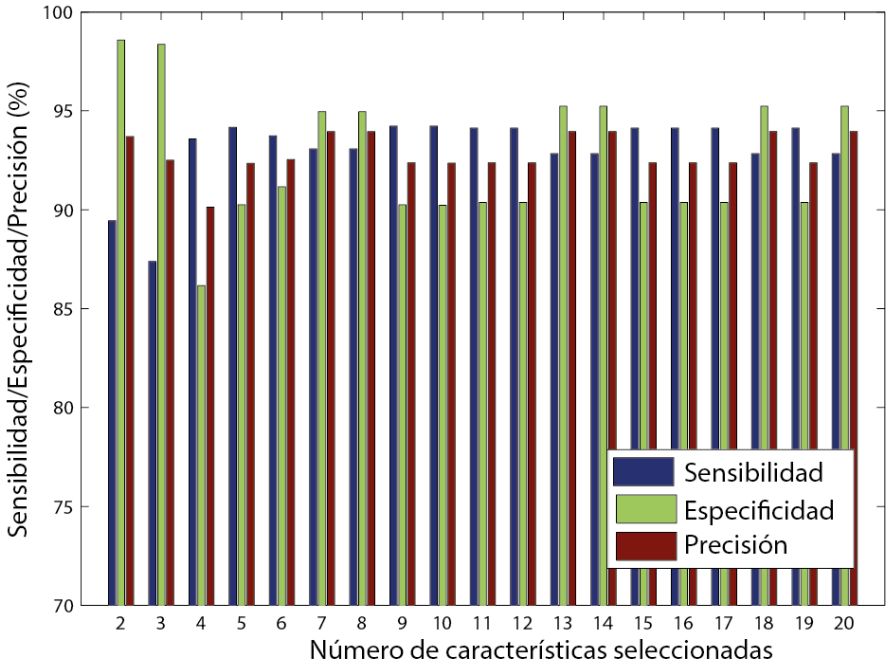


Figura.7.7. Resultados de la clasificación usando PCA+FDR en función del número de características seleccionadas.

Teniendo en cuenta los resultados anteriores se procedió al segundo grupo de experimentaciones de esta primera fase. Para este segundo grupo, se procedió a combinar cada una de las técnicas implementadas hasta ahora con PSOM. Los resultados de estas ejecuciones se exponen en las Figuras. 7.8, 7.9 y 7.10 respectivamente. A partir de estas figuras, el número de características que proporcionan el mejor rendimiento en términos de sensibilidad, especificidad y exactitud se puede calcular como el número más bajo que permite un nivel de rendimiento lo suficientemente alto para cada medida.

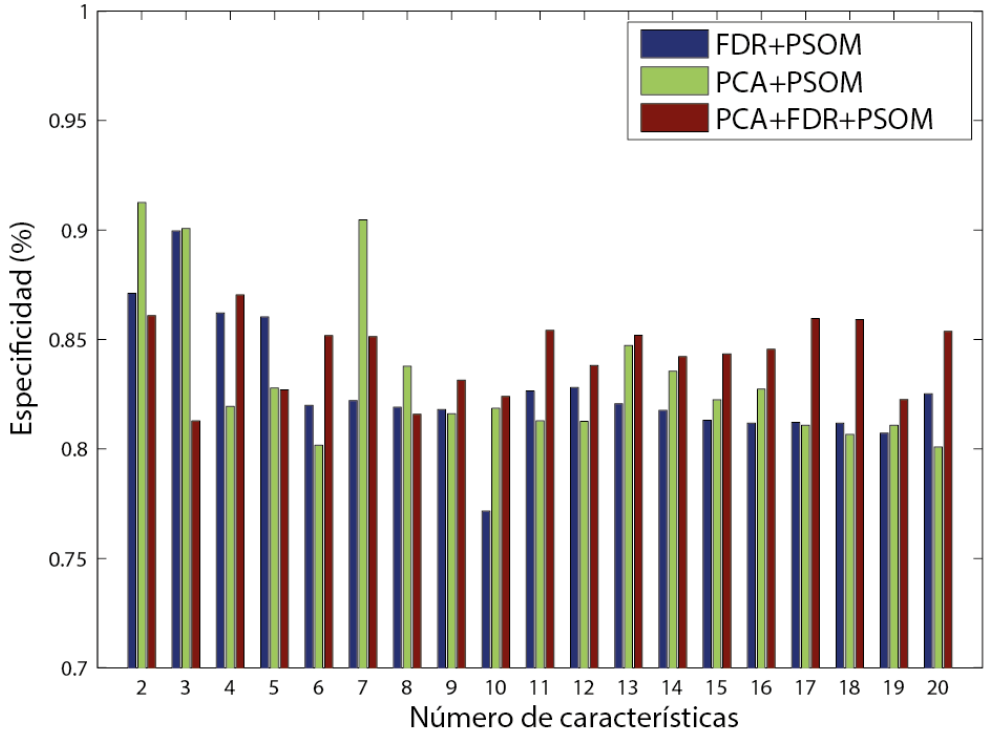


Figura.7.8. Resultados de la clasificación en función del número de características seleccionadas. (Especificidad).

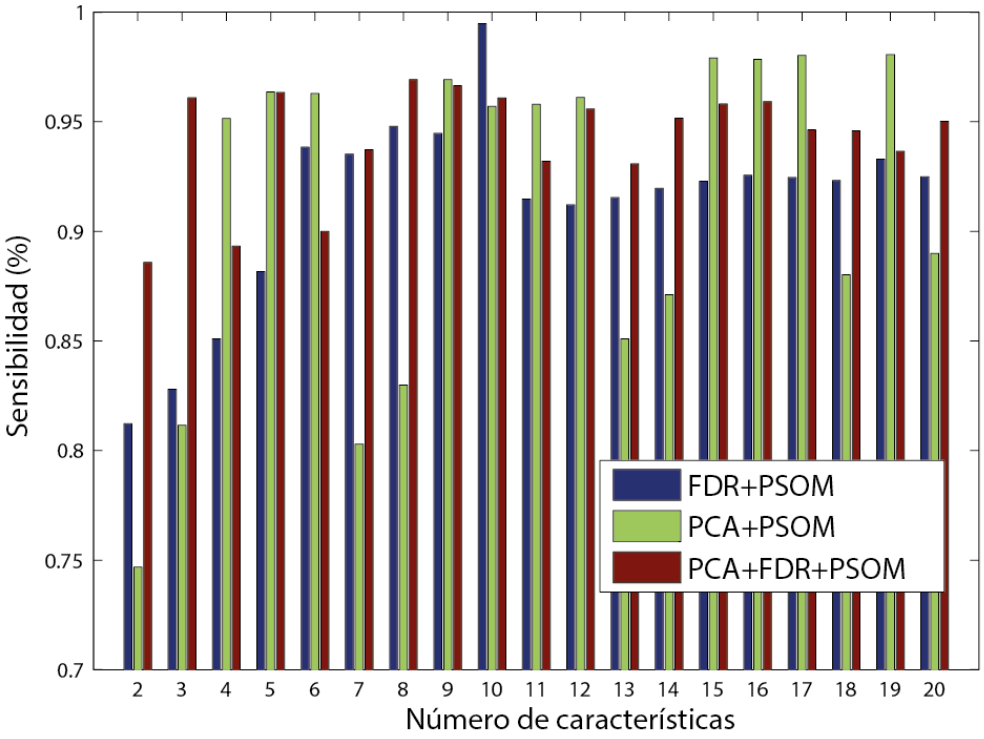


Figura.7.9. Resultados de la clasificación en función del número de características seleccionadas. (Sensibilidad).

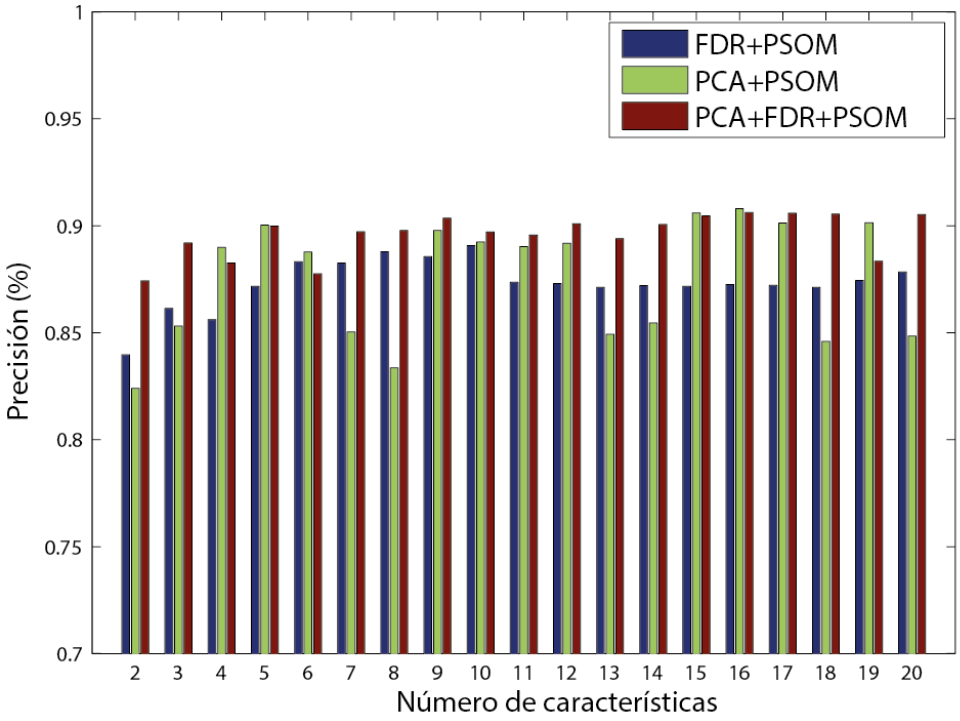


Figura.7.10. Resultados de la clasificación en función del número de características seleccionadas. (Precisión).

En la Tabla 7.5 se muestran los mejores resultados obtenidos para los diferentes métodos de clasificación implementados. El método propuesto PSOM-PCA+FDR es el que proporciona mejores resultados y es indicado en negrita. La desviación estándar para las particiones del conjunto de datos NSL-KDD se calcula a lo largo de 50 ejecuciones. La desviación estándar de la validación cruzada utilizando el conjunto de datos de entrenamiento/test se calcula a lo largo de 10 pliegues y 50 ejecuciones por pliegue.

Esta cantidad de ejecuciones se realizan con el fin de demostrar que el sistema no se sobreajusta y, por lo tanto, tiene un buen rendimiento para la generalización, la selección de características y el entrenamiento. Teniendo en cuenta que con $k = 10$, se hacen particiones aleatorias del 90% de las muestras utilizadas para ajustar el modelo y el resto de las muestras (10%) para la prueba. Estos subconjuntos son disyuntos y no comparten ninguna muestra. Este proceso se repitió 10 veces (tenemos 10 pliegues, es decir $k - fold$ con $k = 10$, asegurándose de que los datos de prueba nunca se utilicen para la selección de características o en el entrenamiento del clasificador. Por lo tanto, los resultados proporcionados por los subconjuntos de características seleccionadas y la precisión de la clasificación se calculan como la media de las 10 evaluaciones correspondiente a las 10 repeticiones (o 10-fold).

Tabla. 7.5. Mejores resultados obtenidos para diferentes métodos de clasificación

Método	No. de características	Precisión	Sensibilidad	Especificidad
NSL-KDD train/test partition				
PSOM+Relief	15	0.87 ± 0.03	0.85 ± 0.05	0.91 ± 0.04
PSOM+CMI	15	0.87 ± 0.02	0.85 ± 0.40	0.93 ± 0.05
PSOM+FDR	9	0.89 ± 0.05	0.98 ± 0.06	0.77 ± 0.06
PSOM+PCA	15	0.90 ± 0.05	0.97 ± 0.05	0.80 ± 0.08
PSOM+PCA+FDR	**8**	**0.90 ± 0.05**	**0.97 ± 0.05**	**0.93 ± 0.06**
10-fold cross-validation (merged dataset)				
PSOM+Relief	15	0.91 ± 0.03	0.90 ± 0.03	0.94 ± 0.02
PSOM+CMI	15	0.90 ± 0.02	0.90 ± 0.04	0.93 ± 0.03
PSOM+FDR	9	0.88 ± 0.01	0.91 ± 0.01	0.90 ± 0.01
PSOM+PCA	15	0.92 ± 0.01	0.88 ± 0.03	0.94 ± 0.02
PSOM+PCA+FDR	**8**	**0.93 ± 0.01**	**0.89 ± 0.02**	**0.96 ± 0.05**

El cálculo implementado en estos estos experimentos se basa en el concepto de validación cruzada $k-fold$ explicado en la Sección 6.3, donde un conjunto de datos D se divide aleatoriamente en k subconjuntos mutuamente excluyentes (los pliegues) D_1, D_2, ..., D_k de aproximadamente el mismo tamaño, como se especificó en la Sección 6.3. El clasificador es entrenado y probado k veces. En cada pliegue $t \in \{1,2,\dots,k\}$, se entrena con el conjunto de datos entre cada pliegue de dicho conjunto (D/D_t) y probado con D_t. La estimación de validación cruzada de precisión vendrá dada por el número total de clasificaciones correctas, dividido por el número de instancias en el conjunto. Formalmente, sea $D_{(i)}$ el conjunto de prueba que incluye $x_i = \langle v_i, y_i \rangle$, entonces la estimación de validación cruzada de la precisión es:

$$acc_{CV} = \frac{1}{n} \sum_{\langle v_i, y_i \rangle \in D} \delta(\mathcal{I}(D \setminus D_{(i),} v_i), y_i) \qquad (7.1)$$

Con $\mathcal{I}(D, v)$ se indica denotara la etiqueta asignada a una instancia v no etiquetada por el clasificador $\mathcal{I}$ en el conjunto de datos D, es decir, $\mathcal{I}(D, v)) = (\mathcal{I}(D))(v)$.

La estimación de validación cruzada es un numero aleatorio que depende de la división en pliegues. La validación cruzada completa es el promedio de todas las $\binom{m}{m/k}$ posibilidades para escoger m/k instancias de m, pero por lo general esto es demasiado costoso.

El número de características indicadas en la Tabla 7.5 son características calculadas en función a los métodos de selección implementados. El número óptimo de características se obtiene a partir del concepto de no-dominancia aplicado sobre los resultados de clasificación [176]. Por ejemplo, la representación de la precisión frente a la sensibilidad. Esta representación para los métodos considerados se muestra en la Figura. 7.11.

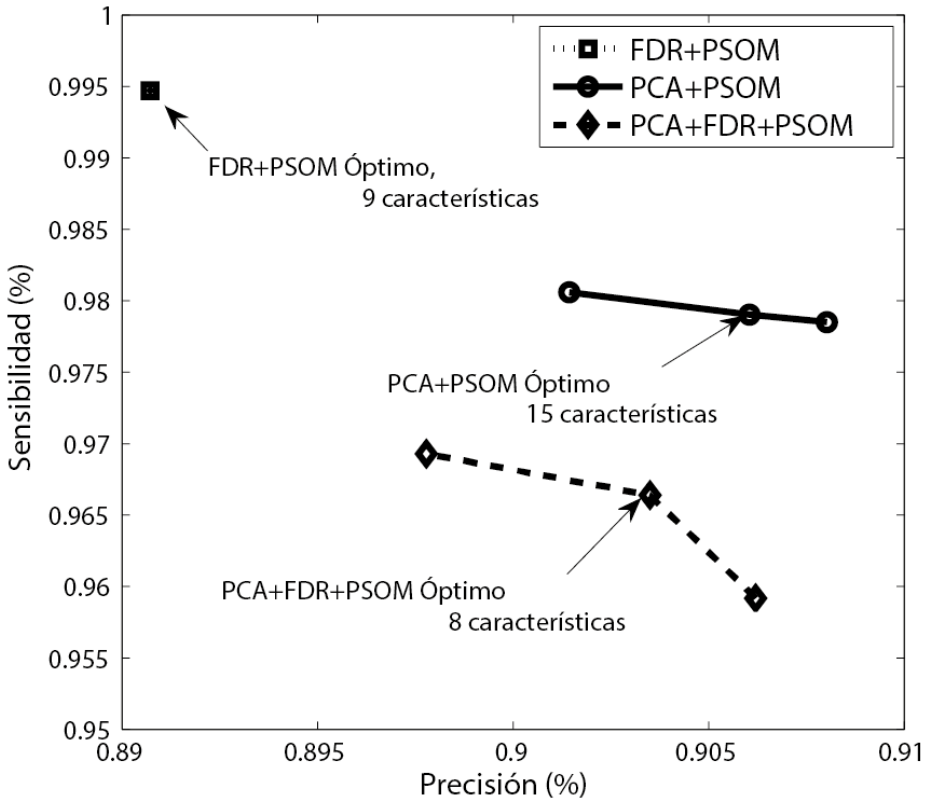


Figura. 7.11. Conjuntos no dominados por diferentes métodos de extracción de características. Los puntos óptimos se indican en cada grafico

En la Figura 7.11, se muestra que el máximo rendimiento de clasificación se consigue con el método propuesto PCA+PSOM+FDR en términos de la precisión y la sensibilidad, utilizando las 15 características obtenidas mediante el método PSOM+PCA. El método PSOM+PCA+FDR ofrece un rendimiento similar con 8 características, pero a medida que se disminuye el número de características se tiene un alto coste computacional. El método PSOM+PCA+FDR es considerado como el que tiene el mejor rendimiento.

El rendimiento de estos tres métodos en términos de precisión, sensibilidad y especificidad se puede comparar estadísticamente para mostrar las diferencias en los valores medios. Sin embargo, no es posible afirmar una clara superioridad de alguno de ellos, tras las pruebas de hipótesis que se han hecho a través de ANOVA [177].

Por otro lado, los resultados obtenidos con otros métodos estadísticos, como el método de *Relief* [178] o algoritmos basados en la teoría de la información, como es el *Conditional Mutual Information (CMI)* [179], han sido incluidos en la Tabla 7.5 para la comparación. Vale la pena mencionar que el número de características indicada en la Figura. 7.11 y en la Tabla

7.5 corresponden a las características específicas seleccionadas dependiendo del método.

En consecuencia, las nueve características seleccionadas por el método FDR+PSOM se refieren a las que proporcionan el valor más alto de FDR (es decir, FDR se ha utilizado como criterio de selección en este método). En el método PSOM+PCA, las características se generan mediante el uso de los vectores propios también llamados autovectores en orden creciente de la varianza explicada y el método FDR+PCA+PSOM selecciona los autovectores cuya proyección proporciona un valor más alto de FDR (es decir, los autovectores más discriminantes de acuerdo con el criterio FDR).

El segundo experimento realizado tiene como objetivo determinar el número de unidades de SOM que maximizan su rendimiento. El número de unidades SOM juega un papel importante en el error de cuantificación y determina la calidad del proceso de auto-organización. Además, la distancia entre las unidades que pertenecen a diferentes clases también depende del tamaño del mapa. Por lo tanto, las *Receiver Operating Curves* (ROC) [180] se han calculado midiendo la diferencia entre la activación media de probabilidad para las unidades etiquetadas como normal o anómalas, mientras se varia el tamaño del mapa.

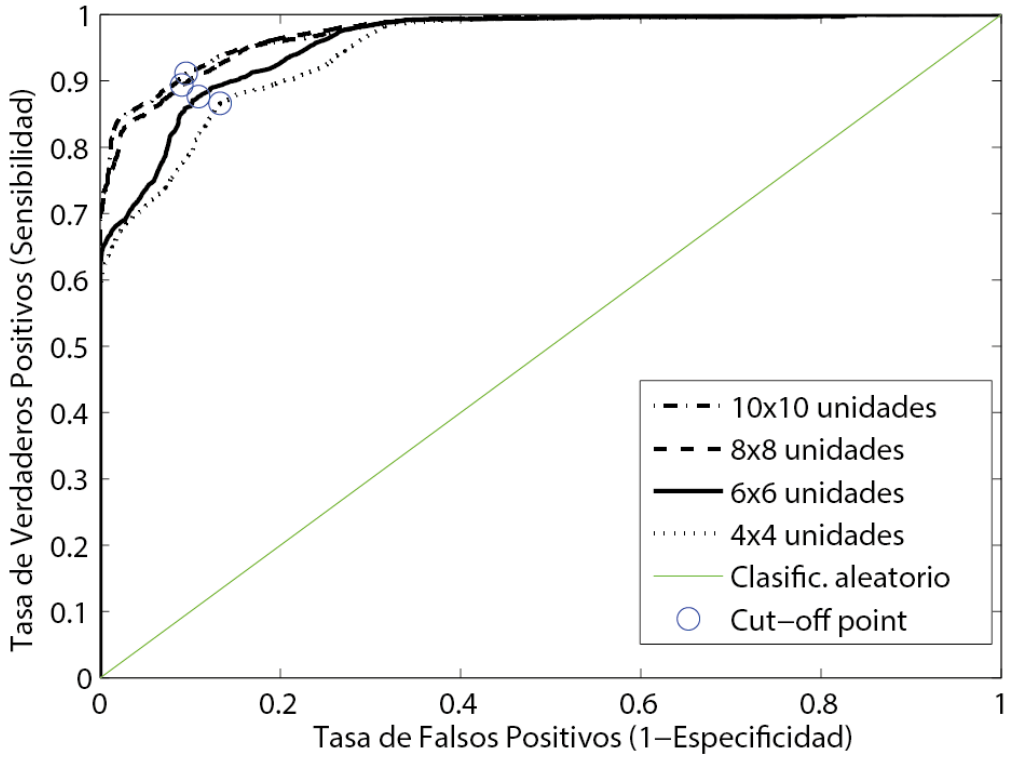


Figura. 7.12. Curvas ROC para los diferentes tamaños de los mapas para PSOM + PCA + FDR.

Las curvas ROC se muestran en la Figura. 7.12, donde se comprueba que el rendimiento máximo del mapa se da para el tamaño del mapa de 8x8 unidades (es decir, el área máxima bajo la curva ROC). En este caso, el punto de corte proporciona valores de sensibilidad de 0,92 y valores de especificidad de 0.95. La Tabla 7.6 muestra el área bajo curva ROC (*AUC, Area Under Curve*) para los diferentes tamaños del mapa. Para el caso puntual de este trabajo de investigación el AUC se ha obtenido para un nivel de significancia del 5%, es decir ($p < 0.05$) lo cual asegura que el tamaño del mapa 8x8 sea mejor que los demás casos que se muestran en la Tabla 7.6.

Tabla 7.6. El área bajo la curva ROC para diferentes tamaños de los mapas para PSOM+PCA+FDR

Tamaño del Mapa	AUC
4x4	0,94
6x6	0,96
8x8	**0,97**
10x10	0,95

A continuación, la Figura 7.13 muestra las probabilidades logarítmicas medias a posteriori para activación para cada unidad de la SOM, tanto para muestras normales, como para las anómalas. El cálculo de dichas probabilidades se hizo a partir de 100 muestras normales y anómalas, y se observa que las muestras normales activan la mayor parte de las unidades normales (círculos azules), manteniendo desactivadas las unidades que representan anomalías (cuadros rojos).

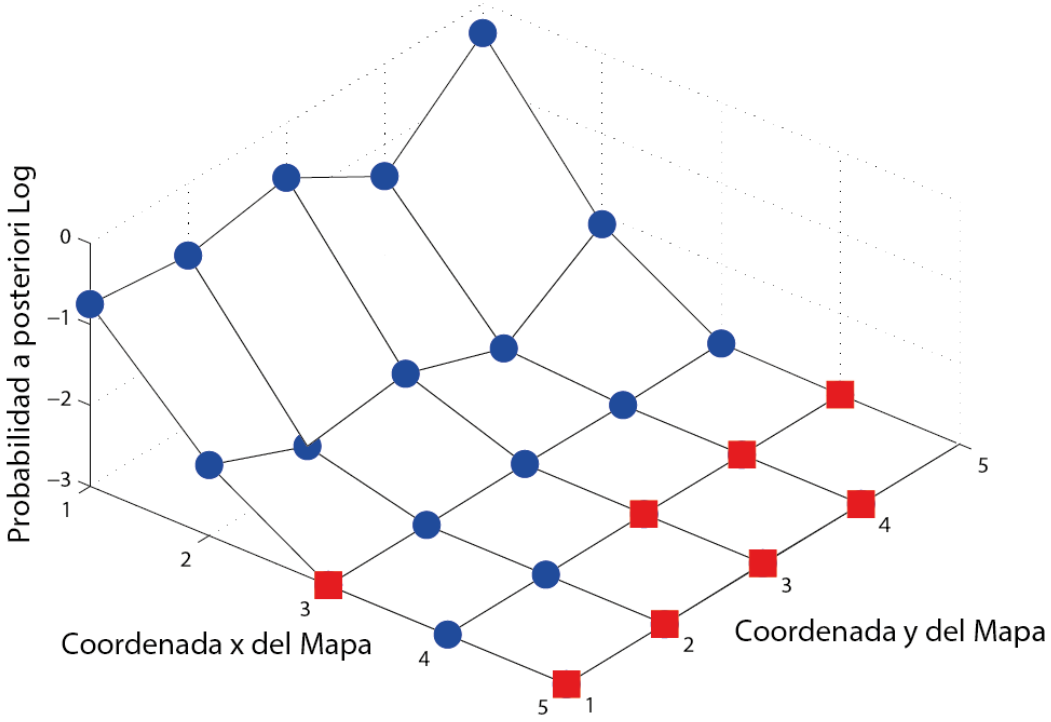


Figura. 7.13. Activación del Mapa SOM para muestras normales.

De manera análoga, en la Figura 7.14 se observa que las muestras anómalas activaron las unidades que representan anomalías (cuadros rojos) mientras que la probabilidad para las unidades normales (círculos azules) decreció notoriamente en comparación con la figura 7.13.

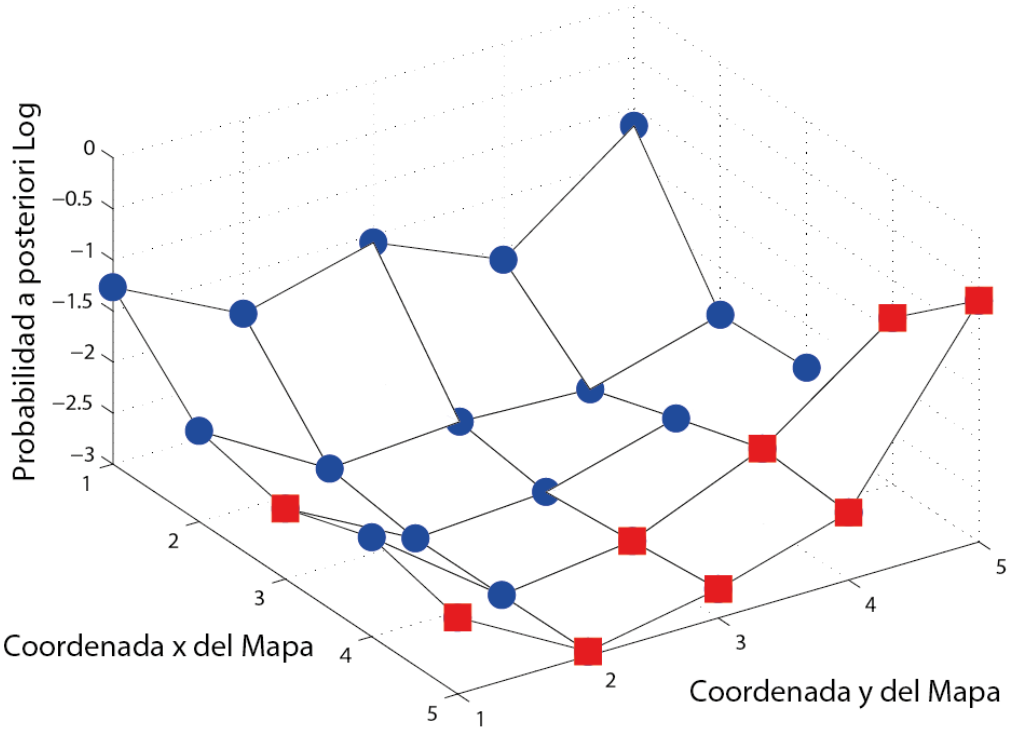


Figura. 7.14. Activación del Mapa SOM para muestras anómalas.

A continuación, se observa la activación del mapa para ambas muestras normales y anómalas. De este modo, la Figura 7.15, muestra el registro de las probabilidades de activación de las unidades de SOM para muestras normales y anómalas en la columna de la izquierda y derecha, respectivamente. Las cifras corresponden a un SOM entrenado de 8x8 con ocho características, tal como se indicó en la Tabla 7.6. Los niveles de activación indican probabilidades posteriores logarítmicas con diferentes colores de acuerdo a la barra de color.

Como se muestra en estas figuras, los patrones de activación para las conexiones normales y anómalas pueden obtenerse a partir de las probabilidades de unidades de mapas similares. Por otra parte, las muestras normales activan la mayoría de las unidades (zonas azules), mientras que las unidades anómalas permanecen inactivas (resto de colores).

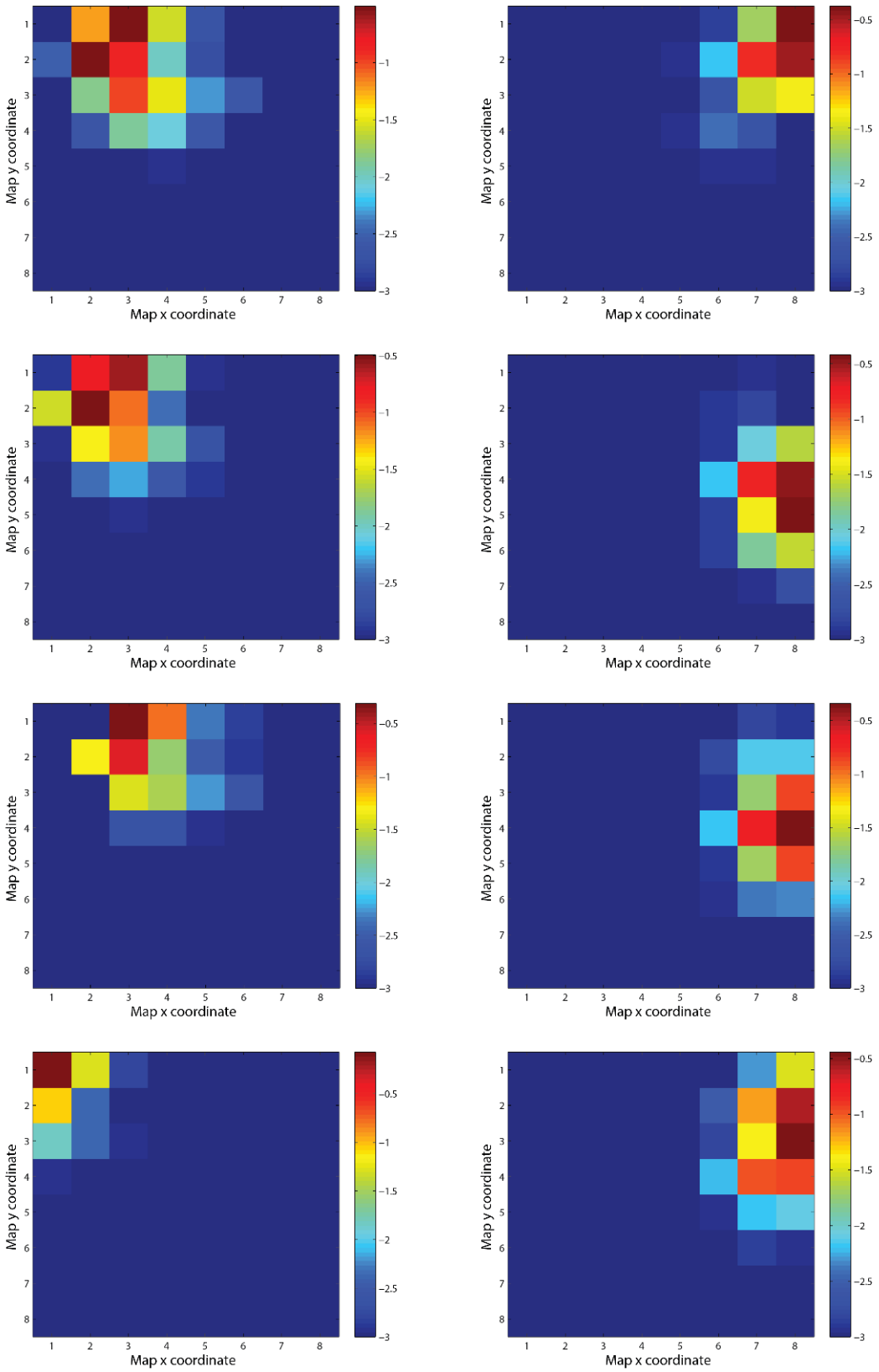


Figura 7.15. Patrones de activación para conexiones normales (columna izquierda) y para anómalas (derecha)

Los resultados de la clasificación con el SOM entrenado utilizando los parámetros anteriormente indicados se aportan en la Tabla 7.5 para la comparación con los mejores resultados obtenidos con otras técnicas de selección de características. Además, la Tabla 7.7 muestra los resultados obtenidos usando todas las características comparando distintos enfoques propuestos. En esta tabla, los resultados de PSOM utilizando PCA+FDR y PCA como métodos de selección de características son los mismos puesto que se utilizan todas las características (es decir, los autovectores no se filtran).

Tabla 7.7 Resultados de la clasificación para los diferentes métodos de clasificación. Se utilizan todas las características en todos los casos para la comparación.

Método	No. De características	Accuracy	Sensitivity	Specificity
Naïve Bayes [181]	41	0,76±*	*	*
Random Forest [181]	41	0,80±*	*	*
DecisiTrees [181]	41	0,81±*	*	*
PSOM+FDR	41	0,60±0,01	0,45±0,30	0,9±0,3
PSOM+PCA	41	0,88±0,01	0,92±0,02	0,84±0,01
PSOM+PCA+FDR	**41**	**0,88±0,01**	**0,92±0,02**	**0,84±0,01**

*No se proporcionan datos por la fuente consultada.

7.3 Conclusiones

A partir de los distintos experimentos realizados con el conjunto de datos, se ha llegado a las siguientes conclusiones:

- El proceso de filtrado y selección de características, así como el entrenamiento y modelado con PSOM que se han propuesto, aumenta la tasa de detección de la mayoría de los ataques (anomalías correctamente identificadas, es decir, verdaderos positivos).

- En cuanto a la tasa de detección según el tipo de ataque (tráfico normal, ataque DoS, ataque Probe, ataque U2R y ataque R2L), el método mixto de clasificación propuesto aumenta el rendimiento de la clasificación y la tasa de detección, en la mayoría de los casos. Exceptuando la categoría de ataque U2R, dado que este tipo de ataque tiene un reducido número de patrones de entrenamiento (las muestras U2R representan sólo el 0,001% de las muestras de

entrenamiento) en el dataset NSL-KDD.

- El rendimiento del sistema sin un proceso previo de filtrado PCA como se observó en las pruebas preliminares disminuye la capacidad de clasificación óptima. Cuando se aplica el método combinado con las demás técnicas expuestas, la tasa de clasificación óptima aumenta.

Dado que se pretende conseguir un IDS/IPS que funcione en tiempo real y pueda detectar comportamientos anómalos y decidir si bloquea o no una conexión, es necesario reducir la complejidad tanto de la selección de características como de la clasificación. Por ello se ha intentado reducir la dimensión del espacio de características optimizando tanto la tasa de clasificación como el tiempo de cómputo y preservar el rendimiento del proceso de etiquetado. Así, el clasificador ha sido entrenado usando vectores con menos dimensiones, permitiendo un cálculo más rápido de las BMU. Por lo tanto, el uso de un menor número de características reduce la carga computacional asociada con el entrenamiento y clasificación de nuevas muestras.

A partir de los experimentos realizados con un conjunto reducido de características 8 y un número de 50 ejecuciones se obtuvo una precisión (*accuracy*) del 90.00%, una sensibilidad (*sensitivity*) del 97.00%, y una especificidad (*specificity*) de 93.00% para los cuatro diferentes tipos de ataques y las conexiones normales usando nuestra propuesta. De manera similar, y usando validación cruzada (*k-fold* con *n=10)* con un número de 50 ejecuciones, los resultados son de una precisión (*accuracy*) del 93.00%, una sensibilidad (*sensitivity*) del 89.00%, y una especificidad (*specificity*) de 96.00% para los cuatro diferentes tipos de ataques y las conexiones normales usando nuestra propuesta. Estos resultados sitúan a nuestra propuesta por encima de los demás métodos encontrados en la literatura.

Capítulo 8. Aportaciones y conclusiones

En este capítulo se describen las conclusiones y aportaciones a las cuales se llega como consecuencia del análisis de los resultados de los experimentos descritos anteriormente. El capítulo finaliza con la identificación de algunos trabajos futuros, que darán continuidad a la propuesta aquí enunciada.

8.1. Aportaciones

El uso de las probabilidades de activación calculadas durante la etapa de entrenamiento, dio como resultado valores de sensibilidad, especificidad y exactitud de hasta el 97 por ciento, 93 por ciento y 90 por ciento, respectivamente. En el método propuesto, las capacidades de clasificación se pueden modificar mediante la variación de las probabilidades de activación previa de las unidades del SOM, evitando la formación del SOM para los nuevos datos. De esta manera, la precisión se puede mejorar mediante el ajuste del umbral de detección. Aunque estas probabilidades previas pueden ser modificados por el administrador de la red, también es posible ajustarlo de forma automática.

La principal ventaja del enfoque propuesto se relaciona con la posibilidad de modificar la clasificación de rendimiento mediante el ajuste de las probabilidades activación previa de cada unidad de SOM, lo que finalmente modifica la clasificación de rendimiento sin re-entrenado. En otras palabras, el sistema puede ser entrenado una vez y se puede ajustar adicionalmente por medio de las probabilidades de activación a priori, evitando la ejecución de todo el proceso de formación de nuevas muestras. Así, nuestras principales contribuciones en este trabajo se pueden resumir de la siguiente manera:

- El uso de un enfoque PCA/FDR para seleccionar las funciones de acuerdo a su poder discriminante.

- El uso de un enfoque hibrido SOM-GMM para modelar patrones normales y anómalos, provee una respuesta difusa de las unidades del mapa que permite medir las probabilidades de activación a posteriori

para nuevas instancias de datos, de acuerdo con el teorema de Bayes.

- La mejora del rendimiento de la clasificación mediante el ajuste de las probabilidades de activación previa, evitando re-entrenamiento cuando nuevos casos disminuyen la precisión de la clasificación, o la sensibilidad.

Además, utilizamos conjuntos estándares y públicos de datos basados en KDD99 y nuestros resultados se han comparados con otros trabajos utilizando el mismo conjunto de datos. Por otra parte, los conjuntos de datos proporcionados se utilizan para tareas de entrenamiento y pruebas respectivamente, asegurando una correcta validación de los resultados [159].

Las aportaciones anteriormente mencionadas, se evidencian como un proceso de investigación realizado a través del desarrollo de esta tesis. En [136] se describen los métodos propuestos en esta tesis los cuales dieron lugar a dicha publicación. Además, en [92] se describe un nuevo e innovador método *wrapper* para selección de características usando optimización multiobjetivo aplicado a la detección de anomalías de red con el uso de GHSOM (*Hierarchical Self-Organising Maps*).

8.2. Conclusiones

En este trabajo se presenta un enfoque PCA/FDR aplicado al problema de detección de intrusiones en redes que se utiliza para la selección de características de acuerdo a su poder discriminante. Además, se ha diseñado un procedimiento hibrido entre las *SOM* bayesianas y un GMM para entrenar el SOM solo una vez, mientras que los resultados de la clasificación se puedan lograr con la modificación de las probabilidades de activación previa de las unidades SOM, de tal manera que permitan que el nivel de activación que se utiliza para reconocer los patrones correspondientes asociados a las conexiones normales y anómalas de red, en cada caso, puedan ser identificados.

En cuanto a los conjuntos de datos utilizados en los procesos de selección, se puso de manifiesto que existen muchas entradas de datos redundantes en KDD'99 y, por tanto, que las técnicas de aprendizaje automático están sesgadas hacia la mayoría de los eventos que ocurren. Esta propiedad conduce a los algoritmos a ignorar eventos menos frecuentes, que pueden ser más perjudiciales que la mayoría de los eventos que ocurren. Los falsos positivos son otro inconveniente importante en el dataset KDD'99. Por ello

en este trabajo se adoptó el conjunto de datos NSL-KDD (una versión mejorada de del conjunto de datos KDD'99).

Los resultados de la clasificación para los diferentes métodos de clasificación evaluados en esta tesis, entregan resultados suficientemente prometedores para el método propuesto. Con el total de las 41 características en todos los casos para la comparación, tanto para el caso del filtrado como para el de la selección de características, se aumentó la detección de verdaderos positivos. Debido a la poca representación de los ataques U2R en el conjunto de datos implementado, el clasificador propuesto no entrega las mejores prestaciones de detección, sin embargo, para el resto de los ataques (DoS, Probe, y R2L) el método mixto propuesto supera el rendimiento tanto de la clasificación como de la detección en la mayoría de los casos con respectos a los demás métodos encontrados en la literatura.

Con el fin de optimizar la tasa de clasificación como el tiempo de cómputo, se redujo la dimensión del espacio de características sin dejar de preservar el proceso del etiquetado. Al reducir la dimensión y usar menos características, el proceso computacional del clasificador disminuye cuando éste se asocia tanto al entrenamiento como a la clasificación de nuevas muestras.

8.3. Trabajos futuros

Se tiene la intención de analizar como el SOM actual podría mejorarse mediante hibridación con otras técnicas de clustering mejoradas, tales como el uso de Maquinas de Soporte Vectorial (SVM). Además, por tanto, como trabajo futuro, se pretende mejorar el cálculo de las probabilidades de activación por medio de optimización multi-objetivo.

Por otro lado, varios SOM podrían combinarse en un conjunto SOM para construir un modelo jerárquico para clasificar no solo las conexiones normales y anómalas, sino también a los cuatro tipos de ataques descritos en el conjunto de datos.

También se analizarán alternativas para implementar eficientemente los sistemas de prevención de intrusos mediante el uso de módulos del núcleo optimizadas en ordenadores con varios procesadores y/o tarjetas de interfaz de red programable (por ejemplo, incluyendo procesadores de red). Consideramos que aprovechando el paralelismo presente en los nodos de

procesamiento actuales mejorará el rendimiento del *IPS*, permitiendo así que los sistemas de prevención de intrusión actúen más eficientemente.

Referencias

[1] J. Gómez, C. Gil, R. Baños, A. López Márquez, F. G. Montoya y M. D. Gil Montoya, «A Pareto-based multi-objective evolutionary algorithm for automatic rule generation in network intrusion detection systems,» Soft Computing, vol. 17, nº 2, pp. 255-263, 2013.

[2] RAE, «Real Academia Española,» 01 04 2015. [En línea]. Available: http://lema.rae.es/drae/?val=seguridad.

[3] RAE, «Real Academia Española,» 01 04 2015. [En línea]. Available: http://lema.rae.es/drae/?val=seguro.

[4] RAE, «Real Academia Española,» 01 04 2015. [En línea]. Available: http://lema.rae.es/drae/?val=informat%C3%ADca.

[5] R. F. Calvo, «ati,» 9 Septiembre 2000. [En línea]. Available: http://www.ati.es/novatica/glosario/glosario_internet.txt.

[6] P. Kotzanikolaou y C. Douligeris, «Computer Network Security: Basic Background and Current Issues,» de Network Security:Current Status and Future Directions, Wiley-IEEE Press, 2007, pp. 1-12.

[7] D. Rusell y G. Gangemi, «Computer Security Basics,» O"Reilly & Associates, Inc., Sebastopol, California, 1991.

[8] NIST, «National Institute of Standards and Technology,» 01 04 2015. [En línea]. Available: http://www.nist.gov/.

[9] D. B. Chapman y D. Zwicky, Construya Firewalls para Internet, Mexico: MacGraw-Hill, 1997.

[10] Z. Lidong y Z. Haas, «Securing ad hoc networks,» Network, IEEE, vol. 13, nº 6, pp. 24-30, 2002.

[11] E. Spafford, «Crisis and Aftermath,» Communications of the ACM, pp. 678-687, 1989.

[12] R. Heady, G. Luger, A. Maccabe y M. Servilla, «The Architecture of a Network Level Intrusion Detection System,» Technical report, Department of Computer Science, University of New Mexico, 1990.

[13] D. Powell y R. Stroud, «Conceptual Model and Architecture, Deliverable D2, Project MAFTIA IST-1999-11583,» IBM Zurich Research Laboratory Research Report RZ 3377, Zurich, 2001.

[14] Fyodor, «Network Mapping Tool,» 01 04 2015. [En línea]. Available: http://www.insecure.org/nmap.

[15] I. f. I. Security, «Institute for Internet Security,» 01 04 2015. [En línea]. Available: http://www.internet-sicherheit.de/en/research/recent-projects/internet-early-warning-

systems/internet-analysis-system/recent-results/ .

[16] G. Fanglu, J. Chen y T. C. Chiueh, «Spoof Detection for Preventing DoSv Attacks against DNS Servers|,» En 26th IEEE International Conference, pp. 37-47, 2006.

[17] K. Sandeep, «Classification and Detection of Computer Intrusions,» citeseer.ist.psu.edu/kumar95classification.html, 1995.

[18] ComputerWire, «DDoS Really, Really Tested UltraDNS. Informe técnico,» 2002. [En línea]. Available: http://www.theregister.co.uk/2002/12/14/ddos_attack_really_really_tested/ attack really really tested.

[19] T. Ylonen, «SSH - Secure Login Connections over the Internet,» En Proceedings of the 6th Security Symposium) (USENIX Association: Berkeley, CA), 1996.

[20] D. C. Feldmeier y P. R. Karn, «UNIX Password Security - Ten Years Later,» citeseer.ist.psu.edu/188968.html, pp. 44-63, 1989.

[21] W. Naiqi, Y. Qian y G. Chen, «A Novel Approach to Trojan Horse Detection by Process Tracing,» Proceedings of the 2006 IEEE International Conference, pp. 721-726, 2006.

[22] J. R. Harrald, S. A. Schmitt y S. Shrestha, «The Effect of Computer Virus Occurrence and Virus Threat Lever on Antivirus Companies,» Engineering Management Conference, IEEE, pp. 780-784, 2004.

[23] D. Brumlen, H. Wang, J. Newsome y D. Song, «Towards Automatic Generation of Vulnerability-based Signatures,» IEEE Symposium, pp. 1081-6011, 2006.

[24] Inteco, «Instituto Nacional de Tecnólogias de la Comunicación,» 01 05 2015. [En línea]. Available: https://www.incibe.es/.

[25] T. Olovsson, «A Structured Approach to Computer Security,» Chalmers University of Technology, 1992.

[26] D. Everett, Identity Verification and Biometrics, Boca Raton, FL, USA: CRC Press, Inc, 1992.

[27] A. V. Huerta, «Seguridad en Unix y redes,» 01 04 2002. [En línea]. Available: https://www.rediris.es/cert/doc/unixsec/unixsec.pdf.

[28] J. Kohl, B. Neuman y T. Ts'o, «The Evolution of the Kerberos Authentication Services,» IEEE Computer Society Press, pp. 79-94, 1994.

[29] B. Daniel, «OSSEC,» 01 04 2006. [En línea]. Available: www.ossec.net.

[30] H. Chet y M. Duren, «Detecting Subtle System Changes Using Digital Signatures,» Information Technology Conference, IEEE, pp. 125-128, 1998.

[31] M. Roesch, «Lightweight Intrusion Detection for Networks,» 2005. [En línea]. Available: www.snort.org.

[32] S. Mbareen, R. B. Vaughn y S. M. Bridges, «Intrusion Sensor Data

Fusion in an Intelligent Intrusion Detection System Architecture,» de Proceedings of the 37th Annual Hawaii International Conference, 2004.

[33] S. X. Wu y W. Banzhaf, «The use of computational intelligence in intrusion detection systems: A review,» Applied Soft Computing, pp. 1-35, 2010.

[34] J. P. Anderson, «Computer Security Threat Monitoring and Surveillance,» James P. Anderson Company, Fort Washington, Pennsylvania, 1980.

[35] O. Dain y R. Cunningham, Fusing Heterogeneous Alert Streams into Scenarios, vol. 6, Springer, 2001, pp. 103-122.

[36] L. Girardin, An Eye on Network Intruder-Administrator Shootouts, Santa Clara, California, 1999.

[37] A. Ghorbani, W. Lu y M. Tavallaee, «Evaluation Criteria. Network Intrusion Detection and Prevention. Concepts and Techniques. Advances in Information Security,» Springer US, pp. 161-183, 2010.

[38] S. X. Wu y W. Banzhaf, «The use of computational intelligence in intrusion detection systems: A review,» Applied Soft Computing, pp. 1-35, 2010.

[39] A. Lazarevic, V. Kumar y J. Srivastava, «Intrusion Detection: A survey,» de Managing Cyber Threats, Minnesota, Springer, 2005, pp. 19-78.

[40] S. Theodoridis y K. Koutroumbas, Pattern Recognition, Burlington : Academic Press - Elsevier , 2009, pp. 1-967.

[41] Computer Security Resource Center, «Computer Security Threat Monitoring and Surveillance,» 15 Abril 1980. [En línea]. Available: http://csrc.nist.gov/publications/history/ande80.pdf.

[42] SRI, «SRI International,» [En línea]. Available: http://www.sri.com/.

[43] T. E. Lunt, «IDES: an intelligent system for detecting intruders,» de Computer Security, Threat and Countermeasures, Rome, 1990.

[44] T. E. Lunt y R. Jagannathan, «A Prototype Real-Time Intrusion-Detection Expert System,» de Security and Privacy, IEEE Symposium, 1988.

[45] H. Debar, M. Dacier y A. Wespi, «A Revised Taxonomy for Intrusion-Detection Systems,» Springer, vol. 55, nº 7-8, pp. 361-378, Julio-Agosto 2000.

[46] S. Axelsson, «Intrusion Detection Systems: A Taxonomy and Survey,» Goteborg, Sweden, 2000.

[47] S. T. Eckmann, «http://citeseerx.ist.psu.edu/,» Department of Computer Science, University of California, 2001. [En línea]. Available:

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.16.4366&rep=rep1&type=pdf.

[48] M. Roesch, «Snort-Lightweight Intrusion Detection for Networks,» Proceedings of LISA '99: 13th Systems Administration Conference, pp. 229-238, 7-12 November 1999.

[49] RAE, «Real Academia Española,» 01 04 2015. [En línea]. Available: http://lema.rae.es/drae/?val=anomal%C3%ADa.

[50] C. C. Wang Ko, Execution Monitoring of Security Critical Programs in a Distributed System: A Specification-Based Approach, 1996.

[51] S. S. Kandeeban y R. S. Rajesh, «Integrated Intrusion Detection System Using Soft Computing,» International Journal of Network Security, pp. 87-92, 2010.

[52] S. B. Sadkhan, «On artificial intelligence approaches for network intrusion detection systems,» MASAUM Journal of Computing, pp. 236-243, 2009.

[53] S. Noel, D. Wijesekera y C. Youman, Modern intrusion detection, data mining, and degrees of attack guilt, George Mason University, 2002.

[54] B. Daniel y J. Sushil, Applications of Data Mining in Computer Security, vol. 6, Springer US, 2002, p. 252.

[55] A. Lazarevic, V. Kumar y J. Srivast, Intrusion Detection: A Survey, vol. 5, Springer US, 2005, pp. 19-78.

[56] J. Buenabad y J. Coria, Tolerancia a fallas para sistemas de detección de intrusos de red, CINVESTAV-IPN, 2004.

[57] S. Northcutt, S. Winters, K. Kent y R. W. Ritchey, Inside Network Perimeter Security: An Analyst Handbook, Second Edition ed., 2005, p. 768.

[58] R. Bace, «An Introduction to Intrusion Detection and Assessment / for System and Network Security Management,» 2000. [En línea]. Available: http://www.iss.net/documents/whitepapers/intrusion.pdf.

[59] SANS, «SANS,» 2015. [En línea]. Available: http://www.sans.org/security-resources/idfaq/.

[60] H. F. Tipton y M. Krause, Information Security Management Handbook, vol. 5, Auerbach Publications, 2006, p. 2036.

[61] K.-R. Muller, A. Smola, G. Ratsch, J. Scholkopf y V. Vapnik, «Using support vector machines for time series prediction».

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

[63] W. Wu, D. L. Massart y S. Jong, «The kernel pca algorithms for wide data part i: Theory and algorithms,» Chemometrics and Intelligent Laboratory Systems, vol. 36, nº 2, pp. 165-172, 1997.

[64] G. Rubio, A. Guillen, L. J. Herrera, H. Pomares y I. Rojas, «Use of specific-to-problem kernel functions for time series modeling,» ESTSP'08: Proceedings of the European Symposium on Time Series Prediction, pp. 177-186, 2008.

[65] D.-Q. Zhang y S.-C. Chen, «Clustering incomplete data using kernel-based fuzzy c-means algorithm.,» Neural Process, vol. 18, nº 3, pp. 155-162, 2003.

[66] E. De la Hoz, A. Ortiz, J. Ortega y E. De la Hoz, «Network Anomaly Classification by Support Vector Classifiers Ensemble and Non-Linear Projection Technique,» de HAIS - Hybrid Artificial Intelligent Systems, Salamanca, España, 2013.

[67] H. Graf, E. Cosatto, L. Bottou, I. Durdanovic y V. Vapnik, «Parallel support vector machines: The Cascade svm,» Advances in Neural Information Processing Systems, pp. 521-528, 2005.

[68] P.-F. Pai y W.-C. Hong, «Support vector machines with simulated annealing algorithms in electricity load forecasting,» Energy Conversion and Management, vol. 46, nº 17, pp. 2669-2688, 2005.

[69] Tay, «Application of support vector machines in financial time series forecasting,» Omega: The International Journal of Management Science, vol. 29, nº 4, pp. 309-317, 2001.

[70] J. R. Schott, «Estimating correlation matrices that have common eigenvectors,» Computational Statistics & Data Analysis, nº 27, pp. 445-459, Diciembre 1998.

[71] S. Theodoridis y K. Koutroumbas, Pattern Recognition, 4th Edition, Elsevier Inc., 2009.

[72] R. O. Duda, P. E. Hart y D. G. Stork, «Pattern Classification and Scene Analysis: Part I Pattern Classification,» de Pattern Classification and Scene Analysis, John Wiley & Sons, 1996.

[73] E. Fix y J. Hodges, «Discriminatory analysis, nonparametric discrimination consistency properties,» 1951.

[74] E. Fix y J. Hodges, «Discriminatory analysis, nonparametric discrimination: small sample performance,» 1952.

[75] T. M. Cover y P. E. Hart, «Nearest neighbor pattern classification,» IEEE Transactions on Information Theory, vol. 13, nº 1, pp. 21-27, 1967.

[76] S. Cost y S. Salzberg, «A weighted nearest neighbor algorithm for learning with symbolic features,» Machine Learning, vol. 10, pp. 57-78, 1993.

[77] T. Kohonen, «The Self-Organizing Map,» Proceedings of the IEEE, vol. 78, nº 9, pp. 1464-1480, 1990.

[78] C. Reeves y G. Singh Billan, «Using Decision Surface Mapping in the Automatic Recognition of Images,» de Artificial Neural Nets and

Genetic Algorithms, Springer Vienna, 2001, pp. 82-85.

[79] D. G. T. Denison, B. K. Mallick y A. F.M. Smith, «A Bayesian CART Algorithm,» Biometrika, vol. 85, nº 2, pp. 363-377, 1998.

[80] H. Chauhan y A. Chauhan, «Implementation of decision tree algorithm c4.5,» International Journal of Scientific and Research Publications, vol. 3, nº 10, 2013.

[81] R. Kaur, G. Kumar y K. Kumar, «A Comparative Study of Feature Selection Techniques for Intrusion Detection,» de 2nd International Conference on Computing for Sustainable Global Development, 2015.

[82] V. Bolón-Canedo, N. Sánchez-Maroño y A. Alonso-Betanzos, «A review of feature selection methods on synthetic data,» Knowledge and Information System, pp. 483-519, 2012.

[83] M. E. Hellman y J. Raviv, «Probability of Error, Equivocation, and the Chernoff Bound,» IEEE Transactions On Information Theory, vol. 16, nº 4, pp. 368-372, Julio 1970.

[84] U. Fano, «Effects of Configuration Interaction on Intensities and Phase Shifts,» Physical Review, vol. 124, nº 6, pp. 1866-1878, 15 Diciembre 1961.

[85] R. P. Duin, «Classifiers in almost empty spaces,» IEEE Explore, 2000.

[86] G. H. John, R. Kohavi y K. Pfleger, «Irrelevant features and the subset selection problem,» de International Conference on Machine Learnig, 1994.

[87] A. Blum y P. Langley, «Selection of relevant features and examples in machine learning,» Artificial Intelligence, pp. 245-271, 1997.

[88] P. Comon, «Independent component analysis, a new concept?,» Signal Process, vol. 36, nº 3, pp. 287-314, 1994.

[89] J. Shlens, A Tutorial on Principal Component Analysis, Center for Neural Science, NYU y Systems Neurology Laboratory, Salk Institute for Biological Studies La Jolla, 2009.

[90] R. Kohavi y G. John, «Wrappers for features subset selection,» Artificial Intelligence - Special issue on relevance, pp. 273-324, 1997.

[91] J. Doak, An evaluation of feature-selection methods and their application to computer security, Tech. rep., University of California, Department of Computer Science, 1992.

[92] E. De la Hoz Franco, E. De la Hoz Correa, A. Ortiz Garcia, J. Ortega Lopera y A. Martinez Alvarez, «Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps,» Knowledge-Based Systems, vol. 71, pp. 332-338, 2014.

[93] W. G. 2. o. t. J. C. f. G. i. M. (. 2. I. v. o. m. —. B. a. g. c. a. a. t. VIM, «Bureau International des Poids et Mesures,» 2008. [En línea]. Available: http://www.bipm.org/utils/common/documents/jcgm/JCGM_20

0_2008.pdf. [Último acceso: 26 Junio 2015].

[94] J. J. Hopfield, «Neural networks and physical systems with emergent collective computational abilities,» Proceedingns of the National Academiy of Sciences, vol. 79, nº 8, pp. 2554-2558, 1982.

[95] R. Lotlikar y R. Kothari, «Multilayer perceptron based dimensionality reduction,» Neural Networks, IJCNN '99. International Joint Conference, vol. 3, pp. 1691 - 1695, 1999.

[96] S. D. Sapkal, S. N. Kakarwal y P. S. Revankar, «Analysis of Classification by Supervised and Unsupervised Learning,» Conference on Computational Intelligence and Multimedia Applications, vol. 1, pp. 280 - 284, 13-15 December 2007.

[97] Y. Liu, «A hybrid neural network learning system,» Computer and Information Technology, 2004. CIT '04, pp. 1016 - 1021, 14-16 September 2004.

[98] T. Kohonen, «Self-organized formation of topologically correct feature maps,» Biological Cybernetics, vol. 43, nº 1, pp. 59-69|, 1982.

[99] G. A. Carpenter y S. Grossberg, «The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network,» Computer, vol. 21, nº 3, pp. 77-88, 1988.

[100] E. Alhoniemi, J. Himberg y J. Vesanto, «Probabilistic measures for responses of self-organizing map units,» Proceedings of the International ICSC Congress on Computational Intelligence Methods and Applications (CIMA), vol. 1, pp. 286-290, 1999.

[101] M. Dittenbach, D. Merkel y A. Rauber, «The growing hierarchical self-organizing map,» Proceedings of the international joint conference on neural networks, vol. VI, pp. 15-19, 2000.

[102] B. Fritzke, «A growing neural gas network learns topologies,» Advances in Neural Information Processing Systems 7, pp. 625-632, 1995.

[103] J. Blackmore y R. Miikkulainen, «Incremental grid growing: Encoding high-dimensional structure into a two-dimensional feature map,» Proceedings of the International Conference on Neural Networks ICNN93, vol. I, pp. 450-455, 1993.

[104] D. Alahakoon, S. Halgamuge y B. Srinivasan, «A structure adapting feature map for optimal cluster representation,» International Conference on Neural Information Processing ICONIP98, pp. 809-812, 1998.

[105] A. Ocsa, C. Bedregal y E. Cuadros-Vargas, «DB-GNG: A constructive self-organizing map based on density,» Proceedings of the International Joint Conference on Neural Networks (IJCNN07), pp. 1953-1958, 12-17 Aug. 2007.

[106] J. Hilera González y V. Martínez Hernando, Redes neuronales

artificiales: fundamentos modelos y aplicaciones, Madrid: Alfaomega Ra-Ma, 2000, p. 390.

[107] T. Kohonen, Self-Organizing Maps, 3 ed., vol. 30, Springer-Verlag Berlin Heidelberg, 2001, pp. XX, 502.

[108] F. Schweitzer, Self-Organization of Complex Structures: from individual to collective dynamics, Berlin: CRC Press, 1997.

[109] S. Haykin, Neural networks, 2 ed., Prentice-Hall, 1999.

[110] P. Bradley y U. Fayyad, «Refining initial points for K-Means clustering,» de Proc. 15th International Conf. on Machine Learning, San Francisco, CA, 1998.

[111] J. Pena, J. Lozano y P. Larranaga, «An empirical comparison of four ini-tialization methods for the k-means algorithm,» Pattern Recogn, vol. 20, pp. 1027-1040, 1999.

[112] A. Juan y E. Vidal, «Comparison of Four Initialization Techniques for the K-Medians Clustering Algorithm,» Proc. of Joint IAPR Int. Workshops SSPR 2000 and SPR 2000 of Lecture Notes in Computer Science, vol. 1876, pp. 842-852, 2000.

[113] J. He, M. Lan, C. Tan, S. Sung y H. Low, «Initialization of cluster refine-ment algorithms: A review and comparative study,» Proceedings of International Joint Conference on Neural Networks (IJCNN), 2004.

[114] E. Forgy, «Cluster analysis of multivariate data: efficiency vs interpretability of classifications,» Biom 21, pp. 768-769, 1965.

[115] J. MacQueen, «Some methods for classification and analysis of multivariate observations,» Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, p. 281–297, 1967.

[116] A. Strehl y J. Ghosh, «Cluster ensembles – a knowledge reuse framework for combining partitionings,» Proceedings of AAAI2002, pp. 93-98, 2002.

[117] J. Ghosh, «Multiclassifier systems: Back to the future,» MCS '02: Proceedings of the Third International Workshop on Multiple Classifier Systems, pp. 1-15, 2002.

[118] T. Kohonen, Self-Organizing Maps, Springer, 2001, p. 502.

[119] T. Mu-Chun SU y H. Chang, «Improving the self-organizing feature map algorithm using an efficient initialization scheme,» Tamkang Journal of Science and Engineering, vol. 5, pp. 35-48, 2002.

[120] O. Elemento, Apport de l'analyse en composantes principales pour l'initialisation et la validation de cartes de kohonen, Inria Nancy - Grand Est: INRIA, 1999.

[121] J. Vesanto, J. Himberg, E. Alhoniemi y J. Parhankangas, «SOM Toolbox for Matlab 5,» 2000.

[122] T. Samad y S. Harp, «Self-Organization with Partial Data,»

Network, pp. 205-212, 1992.

[123] M. M. Borghi, M. Maggiolino, . M. L. L. Montagnani y M. Nuccio, «Determinants in the online distribution of digital content: an exploratory analysis,» European Journal for Law and Technology, vol. 3, nº 2, 2012.

[124] K. J. Friston, J. T. Ashburner, S. J. Kiebel, T. E. Nichols y W. D. Penny, Statistical Parametric Mapping: The Analysis of Functional Brain Images, Elsevier, 2007.

[125] S. K. Ziolko, . L. A. Weissfeld, W. E. Klunk, C. A. Mathis, J. A. Hoge, B. J. Lopresti, S. T. DeKosky y J. C. Price, «Evaluation of voxel-based methods for the statistical analysis of PIB PET amyloid imaging studies in Alzheimer's disease,» NeuroImage, vol. 33, nº 1, pp. 94-102, 2006.

[126] P. F. Odgaard y M. V. Wickerhauser, «Karhunen-Loeve (PCA) based detection of multiple oscillations in multiple measurement signals from large-scale process plants,» American Control Conference, pp. 5893 - 5898, 9-13 July 2007.

[127] S. Zargari y D. Voorhis, «Feature Selection in the Corrected KDD-dataset,» de EIDWT '12 Proceedings of the 3rd International Conference on Emerging Intelligent Data and Web Technologies, 2012.

[128] M. Turk y A. Pentland, «Face recognition using eigenfaces,» de Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on, 3-6 Jun 1991.

[129] Y. Bouzida y S. Gombault, «Eigenconnections to intrusion detection,» de 19th IFIP International Information Security Conference (SEC2004), IEEE, Toulouse, France, 2004.

[130] I. Alvarez Illán, Análisis en Componentes de Imágenes Funcionales para la Ayuda al Diagnóstico de la Enfermedad del Alzheimer, Granada, 2009.

[131] M. Dash y H. Liu, «Feature Selection for Classification,» Intelligent Data Analysis, vol. 1, nº 1-4, pp. 131-156, 24 January 1997.

[132] H. G. Kayacık, A. N. Zincir-Heywood y M. I. Heywood, «Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets,» de Proceedings of the 3rd Conference on Privacy, Security and Trust, 2005.

[133] I. Álvarez Illán, J. Manuel Górriz, J. Ramírez, D. Salas GonzáLez, M. M. López, F. Segovia, R. Chaves, M. Gómez Río y C. Puntonet, «18F-FDG PET imaging analysis for computer aided Alzheimer's diagnosis,» Information Sciences, vol. 181, nº 4, pp. 903-916, February 2011.

[134] M. M. López, J. Ramírez, J. M. Górriz, I. Álvarez, D. Salas

González, F. Segovia, R. Chaves, P. Padilla y M. Gómez Río, «Principal component analysis-based techniques and supervised classification schemes for the early detection of Alzheimer's disease,» Neurocomputing, vol. 74, nº 8, pp. 1260-1271, 8 March 2011.

[135] S.-S. Cheng, H.-C. Fu y H.-M. Wang, «Model-Based Clustering by Probabilistic Self-Organizing Maps,» IEEE TRANSACTIONS ON NEURAL NETWORKS, vol. 20, nº 5, pp. 805-826, 2009.

[136] E. De La Hoz, E. De la Hoz, A. E. Ortiz, J. Ortega y B. Prieto, «PCA filtering and probabilistic SOM for network intrusion detection,» Neurocomputing, vol. 164, pp. 71-81, 21 September 2015.

[137] P. A. Bouvrie, . J. C. A. Angulo y J. S. Dehesa, «Entropy and complexity analysis of Dirac-delta-like quantum potentials,» Physica A: Statistical Mechanics and its Applications, vol. 390, nº 11, p. 2215–2228, 1 June 2011.

[138] T. Heskes, «Self-Organizing Maps, Vector Quantization, and Mixture Modeling,» IEEE TRANSACTIONS ON NEURAL NETWORKS, vol. 12, nº 6, pp. 1299 - 1305, 2001.

[139] A. Dempster, N. Lair y D. Rubin, «Maximum likelihood from incomplete data via the EM algorithm,» Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 39, nº 1, pp. 1-38, 1977.

[140] F. Saâdaoui, «Acceleration of the EM algorithm via extrapolation methods: Review, comparison and new methods,» Computational Statistics & Data Analysis, vol. 54, nº 3, pp. 750-766, 2010.

[141] T. Guoliang, N. Kaiwang y T. Ming, «EM-type algorithms for computing restricted MLEs in multivariate normal distributions and multivariate t-distributions,» Computational Statistics and Data Analysis, vol. 52, nº 10, pp. 4768-4778, 15 June 2008.

[142] H. Wang y Z. Hu, «On EM Estimation for Mixture of Multivariate t-Distributions,» Neural Processing Letters, pp. 243-256, 22 October 2009.

[143] J. Vesanto, J. Himberg, E. Alhoniemi y J. Parhankangas, «SOM toolbox,» Helsinki University of Technology, Finland, 2000.

[144] M. Riveiro, F. Johansson, G. Falkman y T. Ziemke, «Supporting maritime situation awareness using self organizing maps and Gaussian mixture models,» de Proceedings of the 2008 Conference on 10th Scandinavian Conference on Artificial Intelligence (SCAI 2008), 2008.

[145] K. Tasdemir, P. Milenov y B. Tapsall, «Topology-based hierarchical clustering of self-organizing maps,» IEEE Trans Neural Netw, vol. 22, nº 3, pp. 474-485, March 2011.

[146] LL-MIT, «Publications,» 2014. [En línea]. Available: http://www.ll.mit.edu/publications/index.html. [Último acceso: 26 June 2015].

[147] University of California, Information and Computer Science, University of California. Irvine, CA 92697-3425., 28 October 1999. [En línea]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

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

[149] University of California, «KDD Cup 1999 Data,» Irvine, 28 October 1999. [En línea]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. [Último acceso: 15 Agost 2015].

[150] K. Kendall, A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems, 1998.

[151] NSL-KDD. [En línea]. Available: http://www.iscx.ca/NSL-KDD/.

[152] U. O. California, «The UCI KDD Archive,» University of California, 1999. [En línea]. Available: http://kdd.ics.uci.edu/databases/kddcup99/task.html.

[153] M. Bhuyan, D. Bhattachayya y J. Kalita, «Network anomaly detection: methods , systems and tools,» IEEE Commun. Surv. Tutor, p. 99, 2013.

[154] A. Ghorbani, W. Lu y M. Tavallae, Network Intrusion Detection and Prevention: Concepts and Techniques, 2009.

[155] C. Kreibich y J. Crowcroft, «Honeycomb-creating intrusion detection signatures using honeypots,» Proceedings of the 2nd Workshop on Hot Topics in Networks (HotNets-II), 2003.

[156] F. Gong, «Deciphering detection techniques: Part ii,» Anomaly-based intrusion detection McAfee Network Security Technologies Group, White paper, vol. 1, pp. 1-10, 2003.

[157] I. Levin, «KDD-99 classifier learning contest, LLSoft´s results overview,» SIGKDD Explorations, vol. 1, nº 2, pp. 67-75, 2000.

[158] B. Pfahringer, «Winning the FDD99 classification cup: bagged-boosting,» SIGKDD Explorations, vol. 1, nº 2, pp. 65-66, 2000.

[159] M. Tavallaee, N. Stakhanova y A. A. Ghorbani, «Toward credible evaluation of anomaly-based intrusion-detection methods,» IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, pp. 516-524, 2010.

[160] H. G. Kayacik, A. N. Zincir-Heywood y M. I. Heywood, «A

hierarchical SOM-based intrusion detection system,» Engineering Applications of Artificial Intelligence, vol. 20, p. 439–451, 4 Junio 2007.

[161] S.-S. Choi, S.-H. Cha y C. C. Tappert, «A survey of binary similarity and distance measures,» Systemics, Cybernetics And Informatics, vol. 8, nº 1, pp. 43-48, 2010.

[162] E. Eskin, A. Arnold, M. Prerau, L. Portnoy y S. Stolfo, «A geometric framework for unsupervised anomaly detection: detecting intru-sions in unlabeled data,» Applications of Data Mining in Computer Security, 2002.

[163] J. R. Cano, F. Herrera y M. Lozano, «Stratification for scaling up evolutionary prototype selection,» Pattern Recognition Letters, vol. 26, nº 7, pp. 953-963, 15 May 2005.

[164] T. Zseby, «Stratification Strategies for Sampling-based Non-intrusive Measurement of One-way Delay,» Proceedings of Passive and Active Measurement Workshop, pp. 171-179, 2003.

[165] S. Fernandes, C. Kamienski, J. Kelner, D. Mariz y D. Sadok, «A stratified traffic sampling methodology for seeing the big picture,» Computer Networks, vol. 52, nº 14, pp. 2677-2689, 9 October 2008.

[166] S. Geisser, Predictive inference: An Introduction, Minnesota: Chapman & Hall, Inc., 1993.

[167] S. Schwartz y K. M. Carpenter, «The right answer for the wrong question: consequences of type III error for public health research,» Am J Public Health, vol. 89, nº 8, p. 1175–1180, August 1999.

[168] P. A. Devyver y J. Kittler, Pattern Recognition: A Statistical Approach, Michigan: Prentice-Hall, 1982, p. 448.

[169] R. Kohavi, «A study of cross-validation and bootstrap for accuracy estimation and model selection,» de Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, San Francisco, 1995.

[170] P. A. Devijver y J. Kittler, Pattern Recognition: A Statistical Approach, Londres: Prentice-Hall, 1982.

[171] P. Refaeilzadeh, L. Tang y H. Lui, k-fold Cross-Validation, Arizona State University, 2008.

[172] S. J. Raudys y A. K. Jain, «Small sample size efects in statistical pattern recognition: recommendations for practitioners,» IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, nº 3, pp. 252-264, Marzo 1992.

[173] L. Breiman, J. Friedman, C. J. Stone y R. Olshen, Classification and Regression Trees (Wadsworth Statistics/Probability), vol. 1, Boca Raton London New York Washington, DC.: Chapman and Hall/CRC; Edición: New Ed (1 de enero de 1984), 1984, p. 368.

[174] R. R. Bouckaert, «Practical bias variance decomposition,» Advances in Artificial Intelligence - LNCS., vol. 5360, pp. 247-257, 2008.

[175] V. Lakshmanan, A. Fritz, T. Smith, K. Hondl y G. Stumpf, «An automated technique to quality control radar reflectivity data,» Journal of applied meteorology and climatology, vol. 46, nº 3, pp. 288-305, 2007.

[176] D. Kalyanmoy, Multi-Objective Optimization Using Evolutionary Algorithms, NY, USA: Wiley, 2001.

[177] W. Navidi, Statistics for Engineers and Scientists 4th Edition, McGraw-Hill Education, 2014, p. 928.

[178] K. Kira y L. A. Rendell, «The feature selection problem: traditional methods and a new algorithm,» de Proceedings of the 2nd Workshop on Hot Topics in Networks (HotNets-II), AAAI Press, Los Angeles, California, USA, 1992.

[179] F. Fleuret, «Fast binary feature selection with conditional mutual information,» Journal of Machine Learning Research, p. 1531–1555, 5 December 2004.

[180] R. Maxion y R. Roberts, «Proper Use of ROC Curves in Intrusion/Anomaly Detection,» 2004.

[181] M. Panda, A. Abraham y M. R. Patra, «Discriminative multinomial naïve Bayes for network intrusion detection,» de 6th Conference on Information Assurance and Security (IAS), 2010.

El doctorando Eduardo Miguel De la Hoz Correa y los directores de la tesis Dr. Andrés Ortiz García y Dr. Julio Ortega Lopera garantizamos, al firmar esta tesis doctoral, que el trabajo ha sido realizado por el doctorando bajo la dirección de los directores de la tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo, se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

Granada, abril 19 de 2016

Dr. Andrés Ortiz García
Director de la Tesis

Fdo.: Andrés Ortiz García

Julio Ortega Lopera
Dr. Director de la Tesis

Fdo.:

Eduardo Miguel De la Hoz Correa
Doctorando

Fdo.: