ESTUDIO COMPARATIVO DE METODOLOGÍAS DE SELECCIÓN DE CARACTERISTICAS EN SISTEMAS DE DETECCIÓN DE INTRUSOS (IDS), BASADO EN ANOMALIAS DE RED

JORGE LUIS DIAZ MARTÍNEZ

UNIVERSIDAD DE LA COSTA

FACULTAD DE INGENIERIA

MAESTRIA EN INGENIERIA CON ENFASIS EN SISTEMAS

BARRANQUILLA COLOMBIA

2016

ESTUDIO COMPARATIVO DE METODOLOGÍAS DE SELECCIÓN DE CARACTERISTICAS EN SISTEMAS DE DETECCIÓN DE INTRUSOS (IDS), BASADO EN ANOMALIAS DE RED

Autor

JORGE LUIS DIAZ MARTINEZ

Trabajo de Investigación presentado para optar por el título de Magister En Ingeniería

Directores

Dr. Eduardo de la Hoz Correa

Mg. Fabio Mendoza Palechor

UNIVERSIDAD DE LA COSTA

FACULTAD DE INGENIERIA

MAESTRIA EN INGENIERIA CON ENFASIS EN SISTEMAS

BARRANQUILLA COLOMBIA

2016

Nota de aceptación

_____

_____

_____

_____

JURADO

_____

JURADO

_____

JURADO

Barranquilla, octubre 2016

Dedicatoria

Primeramente, a Dios y a mi familia que siempre se mostraron dispuestos apoyarme en todo este proceso de crecimiento personal, de preparación y de ánimo ante las adversidades que suelen suceder en el día a día, solo me queda decirles gracias a ellos.

JORGE LUIS DIAZ MARTINEZ

Resumen

En la actualidad las empresas, no importa su clasificación, poseen diferentes tipos de activos tales como maquinarias, dinero en efectivo, vehículos, cuentas por cobrar, entre otras, sin embargo, el activo más importante que algunas veces pasa desapercibido por la alta gerencia y de la administración de las organizaciones es LA INFORMACIÓN, la información es muy importante en una empresa, tanto que el impacto que llegaría a causar, si llegase a desaparecer o peor aún si cayese en manos de la competencia o de personas malintencionadas, sería realmente funesto, causando serios problemas para el manejo de los procesos organizacionales. Según la organización internacional por la normalización (ISO) define riesgo tecnológico (Guías para la gestión de la seguridad de TI/TEC TR 13335-1) [1996]. Para esta investigación vamos a tener en cuenta los riesgos que se corren al momento de salvaguardar la información de cualquier empresa utilizando tecnologías de redes servidores y clientes, teniendo en cuenta que al momento de implementar estas tecnologías existen ciertas herramientas comerciales que ayudan a salvaguardar la información, minimizando los riesgos informáticos y evitando por tanto accesos intrusivos y diversas tipologías de ataques con los que se pretende causar daños a la información, a la infraestructura de la red y a los equipos conectados. Ante estas situaciones existen diferentes tipos de herramientas y técnicas que nos permiten proteger y reducir el riesgo volviendo nuestra empresa menos vulnerable y endureciendo nuestra plataforma de red. El objeto de esta investigación es abordar una propuesta de selección y clasificación de ataques a redes informáticas soportadas en sistemas de detención y prevención de intrusos IDS/IPS.

Palabras clave: Selección de características, IDS basados en anomalías, Tasas de detección, técnicas de clasificación.

Abstract

Currently companies do not import their asset classification of asset types of customers, cash, vehicles, accounts receivable, among others, however the most important asset that sometimes passes unbalanced by top management and of the administration of the organizations is THE INFORMATION. The information is very important in a company, so much that the impact that would cause, if the result is a disappeared or worse if it fell into the hands of the competition or of malicious people, is really disastrous, causing serious problems for the management of Organizational processes According to the International Organization for Standardization (ISO) define technological risk [Guidelines for the management of IT security / TEC TR 13335-1] [1996]. For this research, we are going to take into account the risks that are in the moment of safeguarding the information of any company using the technologies of networks of servers and clients, taking into account that the moment of implementing these technologies exist tools that help to safeguard the information, minimizing computer risks and avoiding intrusive access and various typologies of attacks that cause damage to information, network infrastructure and connected equipment. In these situations, there are different types of tools and techniques that protect us and reduce the risk making our company less vulnerable and hardening our network platform. The purpose of this investigation is a selection proposal and the classification of attacks of computer networks supported in systems of detention and prevention of intruders IDS / IPS.

*Keywords :*Selection of characteristics, IDS based on anomalies, Detection rates, classification techniques.

Contenido

Introducción

En el presente trabajo se lograra explicar de manera muy transitoria una introducción correspondiente al trabajo de investigación y con el tema de objeto de estudio, realizando una descripción de la principal motivación que genero el desarrollo de dicha investigación en relación con el estudio comparativo de metodologías de selección e características en sistemas de detección de intrusos (IDS), basado en anomalías de red, culminando con una descripción general de la organización estructural de la memoria.

Problema de investigación

Hoy en día estamos en una evolución constante y el hombre en cada momento se encuentra en constante creación e innovación, lo que hoy en día es actualidad en un periodo de tiempo ya no lo es, teniendo en cuenta lo anterior la informática que procura el tratamiento de la información por medio de los computadores, las redes de computadoras y las tecnologías que permiten la conexión de compañías y usuarios entre diferentes partes y entre distintos países, también han evolucionado y la forma en que las empresas almacenan la información y guardan los datos que son de suma importancia ha venido evolucionando también, todo lo anterior nos demuestra que también cambia las formas de almacenar y guardar la información.

De la misma manera también se han generado problema al administrar un sistema informático ya que existen riesgos y vulnerabilidades presentes en estos sistemas y que se pueden presentar en un escenario normal de trabajo en efecto , cuando ingresamos a una página web sin comprender podemos generar acciones como él envió de malware a la red de trabajo, estas acciones pueden ocasionar perdida o daños de la información, teniendo en cuenta que las empresas cuenta con sistemas como los antivirus que logran evitar que se dañe la información ante la presencia de un malware.

Sin embargo, no todas las veces son eficaces en su totalidad ya que en procesos de detección de ataques de red existen ciertas fallas puesto que cuando la base de datos de virus no se actualiza periódicamente y dado que cada vez crecen de manera exponencial nuevos ataques, entiéndase como ataque cuando existe una vulnerabilidad de cualquier índole y el atacante se aprovecha de esta falla, en ese sentido las vulnerabilidades se pueden presentar en sistemas

informáticos y son propias del mismo, es por ello que esta investigación se centra en parte en estos escenarios propios de ambientes informáticos.

Para poder salvaguardar y proteger el activo más importante de las empresas que es la información existen diferentes técnicas de defensa entre esas encontramos: Cifrado de mensajes (Encriptamiento), Control de Acceso (ACLs), Bloqueo de puertos, Redes virtuales Privadas (VPNs), y cortafuegos (FIREWALL). Los que mencionamos en la lista anterior en orden de importancia en un sistema informático lo más deseable es que todos puedan operar de la mejor manera y de forma óptima, los últimos mencionados delimitan el tráfico de servicios de servicios desconocidos, mediante bloqueo de puertos.

Si bien estos son utilizados para contrarrestar una gran diversidad de ataques en el tráfico que viaja por la red permitidos por el dispositivo, queda un hueco de seguridad desde el lado externo que se le denomina (INTERNET), a diferencia de las técnicas de defensa que se mencionaron anteriormente donde encontramos dispositivos que protegen la red interna le conoce comúnmente como (INTRANET). En cuanto a lo anterior tanto el firewall como las anteriores técnicas mencionadas, no controlan los ataques que se generar dentro de la red (INTRANET). Para este tipo de problemas o para estos escenarios y poder mitigar el riesgo se han desarrollado los Sistemas de Detección de Intrusos (IDS) que identifican el tráfico malicioso para proceder a identificarlos bloquearlos y documentarlos que le puedan servir de ayuda al administrador de la red para una posible acción de defensa contra el atacante.

Los IDS pueden detectar ataques con una metodología basada en firmas (comparando los ataques con una base de datos de firmas o reglas) o con una metodología basada en anomalías (empleando un algoritmo de aprendizaje), los basados en firma se han implementado ampliamente en IDS comerciales y en software libre (Snort, Prelude y Suricata), sin embargo,

presentan la limitante de no detectar ataques nuevos; los que emplean algoritmo de aprendizaje detectan ataques nuevos con cierto porcentaje de exactitud.

Del mismo modo las firmas (o reglas) básicamente están constituidas por cabecera y opciones, esto permite identificar el tráfico. La cabecera de firma contiene el protocolo (TCP, UDP, IP o ICMP), las direcciones y puertos tanto de origen como de destino, el sentido de la comunicación y la acción que se ejecutará si coinciden las características del paquete con las condiciones de la regla.

Para poder identificar con precisión la magnitud del problema y las posibles alternativas de solución, se deben abordar con detalle: la fundamentación referida a los Sistemas de Detección de Intrusos, las características inherentes a los dataset DARPA, las técnicas o algoritmos existentes en relación con la extracción de características y técnicas de entrenamiento y clasificación de datos, tales como Redes Neuronales Artificiales y máquinas de soporte vectorial, entre otras.

Producto de las experimentaciones efectuadas, los investigadores han detectado que una variable que incide directamente en la eficiencia del algoritmo de aprendizaje, es la identificación de las características que se van a evaluar durante la fase de pre-procesamiento, debido a que la escogencia de la totalidad de características o algunas de ellas que no sean las apropiadas, generará largos tiempos de respuesta computacional, incidiendo negativamente en la evaluación final del algoritmo de aprendizaje.

El modelo propuesto se destacan varias fases: Pre-procesamiento o normalización, Selección, Entrenamiento, Clasificación y por último evaluación de métricas.

La selección de características en el análisis predictivo se refiere al proceso de identificación de las pocas variables más importantes o atributos que son esenciales en un modelo para una predicción precisa es decir en el caso de la investigación se analizara los patrones que van en una conexión de red agrupadas en un conjunto de datos sintéticos (DATASET), y a este se le aplicaran técnicas de selección de características para teniendo seleccionadas las más relevantes podamos realizar un proceso de entrenamiento .

Seguidamente en la fase de clasificación se carga otro dataset (DARPA NSL-KDD TEST), diferente del conjunto de datos de entrenamiento, reduciendo las características de la nueva colección de datos usando técnicas de selección, teniendo en cuenta las mismas características seleccionadas en la fase de entrenamiento y, por último, se clasifican los datos, basándose en el mapa generado en el proceso de entrenamiento y en el nuevo subconjunto de datos.

Como resultado de todos los procesos anteriores se realizará un último que es el de evaluación de las métricas de desempeño de sensibilidad, especificidad, precisión, y exactitud las cuales permitieron determinar la eficiencia del modelo propuesto.

El tema de estudio concibe un positivo impacto científico, fundamentando las bases de una futura implementación del modelo propuesto de detección de intrusiones en sistemas de red, en IDS comerciales, lo que posibilitará y favorecerá los procesos de detección y clasificación de tráfico normal y anómalo, de forma no supervisada, suprimiendo la necesidad de una actualización manual de la base de datos de ataques, por parte de un especialista humano.

Objetivos

Objetivo general:

Desarrollar un estudio comparativo de metodologías de selección de características aplicadas a sistemas de detección de intrusos (IDS) basadas en anomalías de red con el objetivo de proponer un modelo funcional que mejore las tasas de detección de ataques en entornos informáticos.

Objetivos específicos:

- Documentar referentes teóricos y prácticos referidos al conjunto de datos utilizados en sistemas de detección de intrusos y a las técnicas de selección de características, aplicadas en IDS.
- Identificar la técnica de selección de característica que tenga el mejor comportamiento bajo el modelo propuesto.
- Desarrollar un modelo funcional que hibride una técnica de selección con el método de entrenamiento de red bayesiana

Mapa del documento

El presente trabajo está estructurado por cinco (6) capítulos, cada uno de los cuales se hace una breve descripción a continuación.

Capitulo uno:

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, teniendo en cuentas los diferentes ataques que pueden existir en un ambiente informático, teniendo en cuenta que existen diferentes tipos de amenazas y la cuantificación de las misma.

Capitulo dos:

Se abordan los temas en relación con: Los Sistemas de Detección de Intrusos (IDS), los diferentes mecanismos de prevención, mecanismo de detección y los mecanismos de recuperación las técnicas de eficiencia de un sistema de detección de intrusos (IDS), Se describe la clasificación de los (IDS) sistemas de detección de anomalías, también describe las fases de preprocesamiento, entrenamiento y clasificación de la información en un sistema de detección de intrusos (IDS).

En este capítulo también se abordarán las temáticas correspondientes a la colección de datos que tiene gran relevancia en esta investigación ya para la etapa de experimentación se realizaron por medio de un *Dataset* (Colección de datos), se describirán las técnicas de selección y extracción en una colección de datos.

Capitulo tres:

En este capítulo se plasman los ejes temáticos que fundamentan la investigación en relación con: las técnicas de selección de características y la clasificación que existente dentro las cuales encontramos de filtrado (FILTER), Incrustadas o empotradas (EMBEDDED) y de envoltura (WRAPPER). Se logrará explicar a nivel de detalle cómo funciona cada una de ellas y como es su estructura y modelo.

Capitulo cuatro:

Se expone la propuesta del modelo funcional en todas sus fases y se describirá cada parte de los procesos para saber que pasa en cada una de ellas.

Capítulo cinco:

Se muestra un estudio detallado de diferentes métodos de selección y clasificación de características seleccionadas en la investigación.

Capitulo seis:

Se plasman las conclusiones a las cuales se ha culminado el producto del desarrollo de la tesis de investigación, en la cual se anuncian los resultados obtenidos, a su vez se plantean los trabajos futuros.

Capítulo 1

Con el propósito de la presente investigación, el cual posteriormente será abordado, se hace necesario documentar los siguientes ejes temáticos: seguridad informática, ataques, tipos de ataques y amenazas en un ambiente informático y los sistemas de seguridad existentes para salvaguardar la información en ambientes informáticos.

1.1 Seguridad informática.

La seguridad informática es un tema que le preocupa a las empresas por tal razón existen diferentes herramientas para servir como medios de defensa y cuidar de cierta forma el activo más importante de una organización como lo son los datos, la información. De igual importancia y es de conocimiento común que el terrorismo, las amenazas, los ataques criminales han causado grandes problemas a nuestra sociedades y las infraestructuras comerciales (Bancos, Tiendas Virtuales, Paginas de impuestos, etc.) se han visto afectadas especialmente con la rápida y la gran escala de migración de la información de los medios tradicionales a plataformas de medios sociales, esto conduce a la necesidad de desarrollar más técnicas de defensas de seguridad informática o ciber seguridad. (Wenli, Xiaolong , Tao, & Hiu, 2014).

Según (Russell & Gangemi, 2006) la Seguridad Informática es el cumplimiento de las premisas de confidencialidad, integridad y disponibilidad en un sistema informático, fundamentándose en una serie de elementos conceptuales que es necesario detallar, para una mayor comprensión de los temas de estudio.

Del mismo modo La seguridad informática también conocida con la sigla (SI), se ha vuelto una necesidad para todos los usuarios de computadores en el mundo sin importar el tipo de trabajo que se realice en un ordenador es necesario implementar algún nivel de (SI), en el caso

de usuarios domésticos por lo menos tienen un antivirus y una herramienta de protección para el acceso a internet. En el caso de una empresa debe contar con equipos y herramientas de defensa para proteger la información que se encuentra en la empresa y que sale por medio del (INTERNET), entre esas encontramos firewall, herramientas para correo y spam, herramientas para encriptar los correos y contraseñas (Duran, Martinez Sanchez, & Sanchez Meraz, 2015).

1.1.1. Fundamentos De La Seguridad Informática.

Al pasar del tiempo, el avance de los medios tecnológicos de comunicación ha provocado el surgimiento de nuevos vectores de ataques y de nuevas modalidades delictivas que han transformado a Internet y las tecnologías informáticas es aspectos sumamente hostiles para cualquier tipo de organización, y persona, que tenga equipos conectados a la Word Wide Web.

A diferencia de lo que sucedía años atrás, donde personas con amplias capacidades y habilidades en el área de la informática dedicaban tiempo investigando estos aspectos con el ánimo de incorporar mayor conocimiento; en la actualidad se ha desvirtuado completamente dando origen a nuevos personajes que utilizan los medios informativos y el conocimiento sobre su funcionamiento como herramientas para delinquir y obtener algún beneficio económico.

La seguridad es definida por la RAE (ASALE, 2016) como “cualidad de seguro”; por otra parte, La seguridad informática se fundamenta en una serie de elementos conceptuales que es necesario detallar para una mayor comprensión de los temas a abordar tales elementos son los mecanismos de seguridad utilizadas en contextos organizativos.

Según (Gómez Vieites, 2006) los mecanismos existentes para protección informática son un conjunto de recursos destinados a lograr que los activos de una organización sean confidenciales, íntegros, consistentes y disponibles a sus usuarios, autentificados por

mecanismos de control de acceso y sujetos a auditoria. Un sistema se considera seguro si cumple con las propiedades de integridad, identificación, control de acceso, no repudio, confidencialidad y disponibilidad de la información.

Un sistema se considera seguro si cumple con las propiedades de integridad, identificación, control de acceso, no repudio, confidencialidad y disponibilidad una de estas propiedades conlleva la implementación de determinados servicios y mecanismos de seguridad las cuales se describirán a continuación:

Integridad: Este principio garantiza la autenticidad y precisión de la información sin importar el momento en que se solicita, es decir una garantía de que los datos no han sido alterados ni destruidos de modo no autorizado.

Confidencialidad: (Tung, n.d.)lo define como "el hecho de que los datos o la información esté únicamente al alcance del conocimiento de las personas, entidades o mecanismos autorizados, en los momentos autorizados y de una manera autorizada".

Relación entre confidencialidad e integridad : El flujo de información se puede controlar para incrementar la seguridad mediante la aplicación de modelos como el de (Bell & LaPadula, 1973) para proveer confidencialidad el modelo BIBA (Biba, 1977) para proveer integridad. Ambos modelos son conservadores y restringen operaciones de lectura y escritura para asegurar que no se pueda comprometer la integridad y la confidencialidad de los datos de un sistema. Por ello, un sistema completamente seguro no sería de gran utilidad ya que sería demasiado restrictivo (Kumar, 1995).

El problema de la confidencialidad se vincula comúnmente con técnicas denominadas "encriptación "y la autenticidad con técnicas denominadas "firma digital "aunque la solución de ambos, en realidad, se reduce a la aplicación de procedimientos criptográficos de encriptación y

des-encriptación. Este mecanismo únicamente limita el acceso a un objeto en el sistema, pero no modela ni restringe qué es lo que un sujeto puede hacer con el objeto en el caso de que tenga acceso a su manipulación (Deming, 1992) .

Disponibilidad: Según (Cestero, 2013) define la disponibilidad como "grado en el que los datos están en el lugar, momento y forma en que es requerido por el usuario autorizado". Situación que se produce cuando se puede acceder a un sistema de información en un periodo tiempo considerado aceptable. La disponibilidad está asociada a la fiabilidad técnica de los componentes del sistema de información.

Autenticación (IDENTIFICACION): El sistema debe ser capaz de verificar que un usuario identificado que accede a un sistema o que genera una determinada información es quien dice ser. Solo cuando un usuario o entidad ha sido autenticado, podrá tener autorización de acceso. Se puede exigir autenticación en la entidad de origen de la información, en la de destino o en ambas. (Piattini & Peso, 2001) .

No repudio o Irrenunciabilidad: Proporcionará al sistema una serie de evidencias irrefutables de la autoría de un hecho. El no repudio consiste en no poder negar haber emitido una información que si se emitido y en no poder negar su recepción cuando si ha sido recibida.

De esto se deduce que el NO REPUDIO puede darse:

En origen: El emisor no puede negar el envío porque el receptor tiene pruebas certificadas del envío y de la identidad del emisor. Las pruebas son emitidas por el propio emisor.

En destino: En este caso es el destinatario quien no puede negar haber recibido el envío ya que el emisor tiene pruebas infalsificables del envió y de la identidad del destinatario. Es el receptor quien crea las pruebas

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.

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. Teniendo en cuenta las definiciones anteriores se enuncian el concepto de seguridad informática según (Russell & Gangemi, 1991), 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.

1.1.2. *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 (NIST, 2015) 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 (Chapman & Zwicky, 1997) y (Lidong & Haas, 2002) 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.

2.1 Ataques y amenazas.

Un Ataque informático consiste en aprovechar alguna debilidad o falla (Vulnerabilidad) en el hardware el software e incluso en las mismas personas que formar parte del ambiente informático; con el objetivo de obtener un beneficio por lo general la mayoría de las veces de índole económico, causándole un efecto negativo en la seguridad del sistema que luego esto repercute en los activos de la empresa u organización.

Para minimizar el impacto negativo provocado por ataques, existen procedimientos y mejores prácticas que facilitan la lucha contra las actividades delictivas y reducen notablemente el campo de acción de los ataques. Uno de los pasos más importantes en seguridad, es la educación. Comprender cuáles son las debilidades más comunes que pueden ser aprovechadas y cuáles son sus riesgos asociados, permitirá conocer de qué manera se ataca un sistema informático ayudando a identificar las debilidades y riesgos para luego desplegar de manera inteligente estrategias de seguridad efectivas.

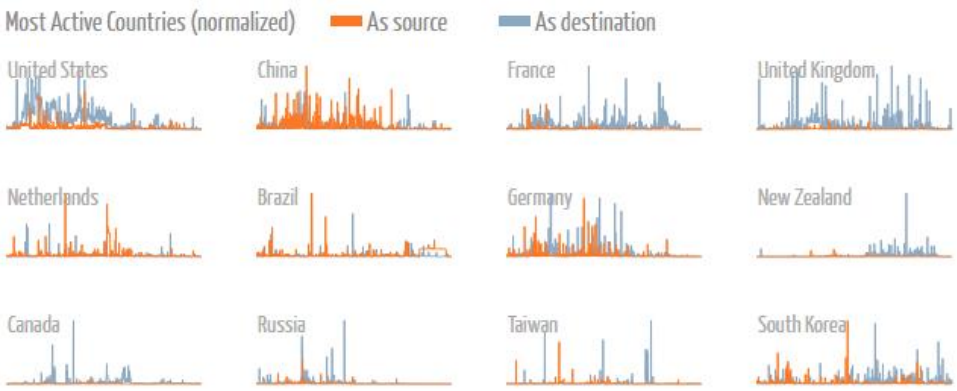

Figura #1.1 Visualización de ataques en Visualización ataques por países 1

Fuente:(Google Ideas, 2000).

En la ***figura 1.1*** nos muestra cómo se están dando los diferentes tipos de ataques en diferentes países del mundo, en el listado se encuentra los más activos el color anaranjado nos muestra la fuente del tipo de ataque en este caso la mayoría de los ataques son de DNS, y el color gris indica el destino hacia dónde va dirigido el ataque esta herramienta es muy importante ya que es un proyecto construido mediante una colaboración de Google Ideas y Arbor Networks, la herramienta superficies de los datos de tráfico de ataque anónimos para que los usuarios exploren las tendencias históricas y encontrar informes de interrupciones que ocurren en un día determinado.

Teniendo en cuenta los riesgos que implican el poder tener conexión en un sistema informático el protocolo de comunicación permite 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* (Spafford, 1989), ha habido una innumerable cantidad de intrusiones de red que se han saltado los mecanismos establecidos para la protección 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 (Heady, Luger, Maccabe, & Servilla, 1990) 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 (Powell & Stroud, 2001), 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.

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 (Fyodor, 2015), 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 1.2*** suministrada por el *Institute for Internet Security* (IIS) (Security, 2015).

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.

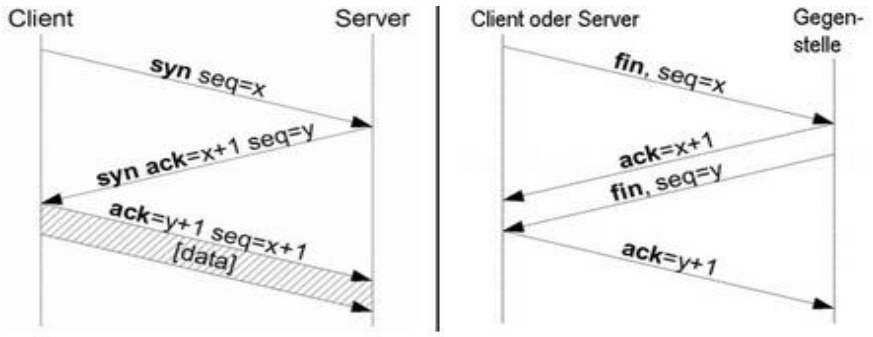

Figura 1.2. TCP Connect y TCP SYN (Security, 2015).

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 (Fanglu, Chen, & Chiueh, 2006) 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 (Sandeep, 1995):

  - 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 (Fanglu, Chen, & Chiueh, 2006) y un ataque de negación de servicio (ComputerWire, 2002).

- 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 (Ylonen, 1996), diccionarios de datos (Feldmeier & Karn, 1989) o programas maliciosos como los troyanos (Naiqi, Qian, & Chen, 2006) y virus informáticos (Harrald, Schmitt, & Shrestha, 2004).

La mayoría de los atacantes aprovechan las vulnerabilidades que existen en los sistemas informáticos. Se trata de errores (Brumlen, Wang, Newsome, & Song, 2006) 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 (Inteco, 2015), 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.

Así mismo debemos tener en cuenta las amenazas presentes en un ambiente informático nos encontramos con técnicas de avanzadas de evasión y por lo general pasan por alto los controles de seguridad e incluso podrían permanecer en el sistema durante mucho tiempo sin ser detectado, y sin cualquier rastro observable, es importante que se tenga en cuenta este panorama de amenazas dinámicos y complejo y lo que afectan a las empresas como lo son las:

- **Amenaza:** Como afirma (Knake & Clark), Una amenaza representa un programa malicioso que es utilizado por individuos llamados ciberdelincuentes, de igual forma en eventos o acciones que violentan la integridad disponibilidad y confidencialidad de la información, que pueden desencadenar un incidente en las personas y/o en la plataforma de una organización, ocasionando pérdidas humanas daños materiales o perdidas inmateriales de sus activos.

- **APT (amenaza persistente avanzada):** Es un conjunto de procesos informáticos con la capacidad de eludir las protecciones de seguridad normales, debido a que utilizan software malicioso para explotar las vulnerabilidades en los sistemas, cualquier malware puede comportarse como una APT, esta técnica es dirigida principalmente a organizaciones o empresas para tratar de robar y filtrar información sin ser identificados (Peter & Friedman, 2014).
- **Tasa de Amenaza:**

  Una taza de amenazas es el número de caso de amenazas observados durante un periodo de tiempo y se calcula dividiendo del total de caso observado por el periodo de tiempo que se considere o el tiempo que dure la observación, como se muestra en la ecuación (1.1).

$$Tasa\ Amenenazas = \frac{Numero\ Amenazas}{Tiempo\ Observado} \quad (1.1)$$

- **Probabilidad De Amenaza:**

  Es la probabilidad de posterior basada en los casos observados de taza de amenaza durante un periodo de tiempo. Este se calcula dividiendo los casos observados de tasa de amenazas por el número total de todas las distintas amenazas sobre un periodo específico de tiempo, como se muestra en la ecuación (1.2).

$$Probabilidad\ de\ Amenaza = \frac{Amenza * Numero\ de\ Amenazas}{Tiempo\ Especifico} \quad (1.2)$$

Capítulo 2

En este capítulo se realizara una muestra detallada de conceptos como lo son los mecanismo de prevención, detección y recuperación, enfatizando los fundamentos teóricos de un sistema de detección de intrusos (IDS), además se introduce definiciones características, eficiencia y clasificación de los sistemas de detección de intrusos haciendo énfasis de cómo es cada uno de ellos y la importancia que tienen estos sistemas al momento de aplicar técnicas de seguridad y protección de información en organizaciones, de igual importancia se describe el funcionamiento, eficacia y modelo de un sistema de detección de intrusos y se describen en este capítulos los conceptos de colección de Datos *(dataset),* y seguidamente las técnicas de selección y extracción de características. De igual forma se logra explicar las técnicas de clasificación que en la cual se basa esta investigación como lo son las redes bayesianas *(Bayes Net).*

2.1 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 (Olovsson, 1992):

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 (Everett, 1992) 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 (Huerta, 2002), cifrados de clave pública, privada, firmas digitales, etc. Aunque cada vez se utilizan más los protocolos seguros como SSH o Kerberos (Kohl, Neuman, & Ts'o, 1994) 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.

2.1.1 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 (Daniel B. , 2006), Tripwire (Chet & Duren, 1998), Snort (Roesch, Lightweight Intrusion Detection for Networks, 2005), RealSecure (Mbareen, Vaughn, & Bridges, 2004), y Prelude (Wu & Banzhaf, 2010), entre otros.

2.1.2. ***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).

2.2 Fundamentos De Sistemas De Detección De Intrusos (IDS).

Los sistemas de detección de intrusos son muy importantes ya que son herramientas de eficaces herramientas de protección de datos que complementan en gran medida el uso de otras técnicas de seguridad. En la búsqueda de mejorar la complejidad de la auditoría y la habilidad para la vigilancia de sistemas informáticos James P. Anderson (Anderson, 1980) 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 (Dain & Cunningham, 2001), 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 (Girardin, 1999), 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.

2.2.1. *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 (Ghorbani, Lu, & Tavallaee, Evaluation Criteria. Network Intrusion Detection and Prevention. Concepts and Techniques. Advances in Information Security, 2010).

La ***Tabla 2.1*** se denomina matriz de confusión y ha sido definida en (Wu & Banzhaf, 2010) y (Lazarevic, Kumar, & Srivastava, Intrusion Detection: A survey, 2005). 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 (Lazarevic, Kumar, & Srivastava, Intrusion Detection: A survey, 2005) y conceptualizada en (Ghorbani, Lu, & Tavallaee, Evaluation Criteria. Network Intrusion Detection and Prevention. Concepts and Techniques. Advances in Information Security, 2010) y (Theodoridis & Koutroumbas, Pattern Recognition, 2009), se aprecia que el

IDS perfecto será aquel que detecte todo el tráfico de forma correcta, no generando ninguna falsa alarma.

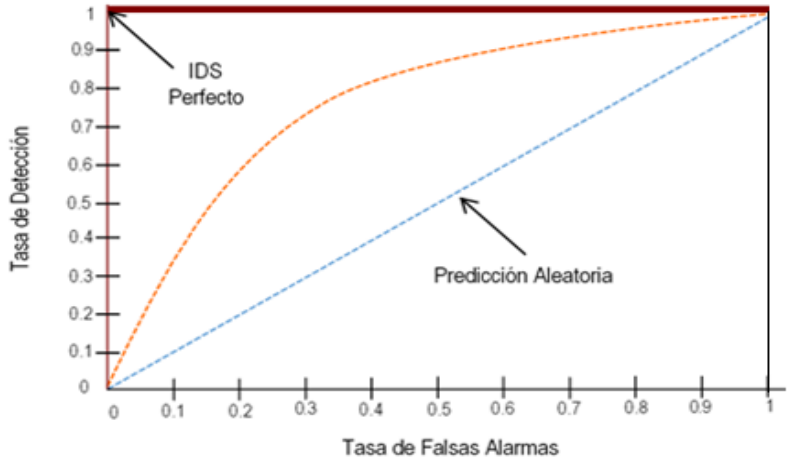

Figura 2.2 Curva ROC

2.2.2. *Clasificación de los Sistemas de Detección de Intrusos.*

La clasificación o taxonomía de los sistemas de detección de intrusos Figura 3.2., ha sido tratada en numerosos trabajos, de los que destacan los de Hervé Debar (Debar, Dacier, & Wespi, 1999) y Stefan Axelsson (Axelsson, 2000) de Chalmers University of Technology en Suecia, los cuales se clasifican de acuerdo con los criterios de enfoque o tipo de análisis, origen de los datos o fuentes de información, por su estructura y según su respuesta o comportamiento (Guttman & Roback, 1995) (Lunt, Tamaru, & Gillham, 1992).

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

Fuentes de Información:

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

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

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

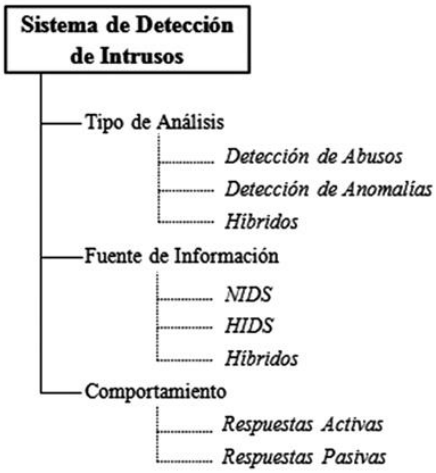

Figura #2.3. Clasificación de los Sistemas de Detección de Intrusos (IDS).

2.2.3. Estrategias de análisis en los Sistemas de Detección de intrusos

Después del proceso de recopilación de información, se lleva a cabo el proceso de análisis. Los dos tipos principales de análisis son, ***Fig. 2.4***:

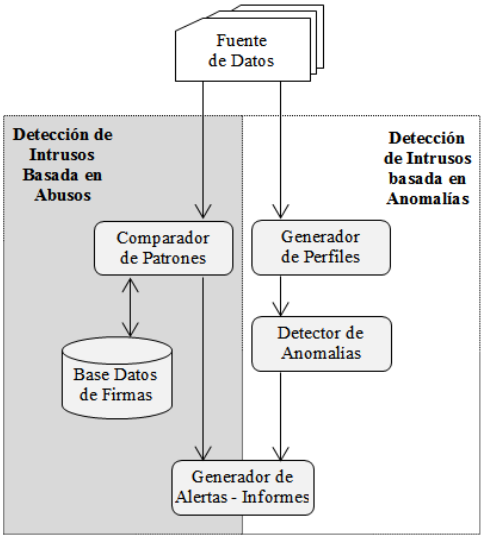

Figura 2.4 Esquema de los IDS de acuerdo con el enfoque

- **Estrategia De Análisis: Detección De Abusos**

Un IDS basado en detección de uso indebido o abusos, monitoriza las actividades que ocurren en un sistema y las compara con firmas de ataques, las cuales se encuentran almacenadas en una base de datos.

Cuando las actividades monitorizadas coinciden con las firmas, generan una alarma. La detección de intrusos basada en uso indebido se atiene al conocimiento a priori de las secuencias y actividades que forman un ataque. Con este método se detectan las tentativas de explotación de vulnerabilidades conocidas o patrones de ataque típicos. Esta estrategia es la más utilizada en los IDS comerciales y por la que apuestan los fabricantes. Típicamente, un sistema de detección de uso indebido contiene dos componentes principales según Kumar (Kumar & Spafford, 1994) :

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

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

Como desventaja, cabe mencionar el hecho de que el patrón del ataque ha de ser conocido con anterioridad, lo que hace que nuevas intrusiones pasen desapercibidas ante el detector, o que el sistema pueda ser fácilmente engañado con pequeñas variantes de los patrones de ataques conocidos. Otra desventaja es que hay que adaptar manualmente el IDS al sistema en el que se implanta si no queremos que se dispare el número de falsos positivos una intrusión anómala, la actividad es no intrusiva, pero como es anómala el sistema decide que es intrusiva. Se denominan falsos positivos, porque el sistema erróneamente indica la existencia de intrusión.

- **Estrategia De Análisis: Detección De Anomalías**

Consiste en la elaboración de perfiles estadísticos de comportamiento a lo largo del tiempo. Estos perfiles se construyen mediante determinados algoritmos, capaces de detectar cambios graduales en los patrones de conducta de los usuarios o anomalías. Una anomalía se puede definir como la discrepancia de una regla o de un uso (Vidal, Vega, & Guijarro, 2012).

De ese modo, el primer paso de un sistema de detección de anomalías comienza por establecer lo que se considera comportamiento normal de un sistema (usuarios, redes,

registros de auditoría, llamadas del sistema de los procesos, etc.). Una vez definido esto, clasificará como sospechosas o intrusivas aquellas desviaciones que pueda detectar sobre el comportamiento normal. La detección de anomalías depende mucho de la suposición de que los usuarios y las redes se comportan de un modo suficientemente regular, de forma que cualquier desviación significante pueda ser considerada como evidencia de una intrusión.

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

Como desventaja, por definición únicamente señala comportamientos inusuales, pero éstos no tienen necesariamente por qué ser ilícitos. Por ello, destaca el problema de su alta tasa de falsos positivos. Otra desventaja de este proceso es la falta de claridad. Un intruso podría actuar lentamente y realizar sus acciones cuidadosamente para modificar el perfil de los usuarios de modo que sus actividades serían aceptadas como legales cuando en realidad deberían lanzar una alarma (falsos negativos).

2.3 Funcionamiento y modelo de simulación en sistemas de detección de intrusos

La eficacia del proceso de detección del tráfico malicioso en una red informática, mediante la aplicación de un IDS que utilice técnicas de reducción de características, algoritmos de aprendizaje de máquina y detección de tráfico anómalo, es susceptible de ser evaluada mediante simulación de laboratorio. Ello requiere de la ejecución de varias fases: escogencia de la colección de datos (dataset), pre-procesamiento, normalización, entrenamiento (training) y clasificación. La Fig. 2.5 ilustra dichas fases.

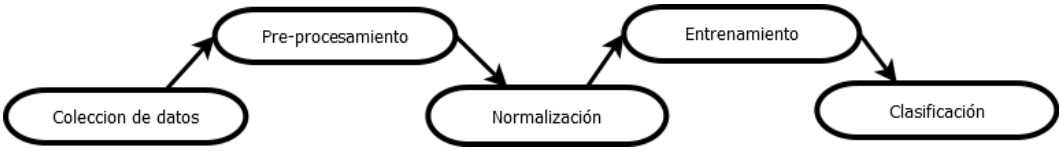

Figura 2.1 Fases del proceso de simulación de intrusos.

Fase de elección de la colección de datos:

En esta fase inicial se debe seleccionar la colección de datos que se va a usar para las fases subsiguientes. Aunque existe una amplia variedad de datasets los investigadores comúnmente se han decantado por el uso de DARPA NSL-KDD, por las ventajas que ofrece en relación con la variedad y depuración de sus datos con respecto a otros dataset de su misma familia y de otras organizaciones.

Fase de preprocesamiento:

Los datos procedentes del dataset deben estar en el rango de [0 a 1] o de [-1 a 1]. Sin embargo, no lo están, debido a que todas las conexiones en sus 41 características poseen valores continuos, discretos o simbólicos y en diferentes rangos de significancia. Con el propósito de estandarizar dichos valores para que puedan ser eficazmente procesados por los algoritmos de aprendizaje de máquina, se debe hacer un preprocesamiento y normalización de los datos contenidos en las conexiones.

Para la conversión de los símbolos en formato numérico, a cada símbolo se asigna un código entero. Por ejemplo, en el caso de la característica protocol_type, se asigna “0” a tcp, “1” a udp y “2” a icmp. De forma similar los nombres de ataque son mapeados asignando valores enteros a las cinco categorías así: “0” para tráfico normal, “1” para el ataque de sondeo (probe), “2” para la Denegación de Servicios (DoS), “3” para U2R y “4” para R2L.

Por otra parte, debido a que existen características cuyos valores se extienden por un rango de números enteros muy grande, es decir, src_bytes toma valores entre [0 y 1.3 billones] igual que dst_bytes. Se aplica entonces una escala logarítmica (de base 10) a estas características para reducir el rango de [0.0 a 9.14]. Todas las demás características son booleanas, en el rango de [0.0 a 1.0]. Por lo tanto, el escalado no es necesario para estos atributos.

En la fase de pre-procesamiento también se debe identificar la técnica de reducción de características que se van a utilizar, debido a que no es conveniente efectuar el entrenamiento de la red con la totalidad de características, dado que ello podría ralentizar considerablemente el procesamiento, sin añadir una significativa exactitud en la clasificación del tráfico. Posteriormente se abordarán las técnicas de reducción de características más utilizadas en la actualidad.

Fase de entrenamiento:

En esta fase se entrena la red neuronal a partir del algoritmo de aprendizaje seleccionado y tomando como insumo el archivo procedente del dataset para tal fin. Normalmente se usa un archivo que contiene una cantidad de registros equivalente al 100% del total de los datos contenidos en el dataset (KDDTrain+).

Fase de clasificación:

Una vez la red neuronal ha sido entrenada, se procede con la fase de clasificación en la cual, de forma autónoma, el algoritmo clasificador determina qué tráfico es normal y cuál es un ataque, efectuando la subsiguiente clasificación de cada una de las conexiones del dataset.

Gracias a esto se podrá presentar la información de resumen del proceso, de forma estadística, mediante gráficos por estado (tráfico normal o ataques), agrupados por tipo de ataque y listando las métricas de desempeño para valorar la eficiencia del sistema. Una vez la red

neuronal está entrenada, se procede con la prueba, la cual se realiza con el 100% de los datos contenidos en el dataset; para ello usualmente se utiliza el dataset kddtest.

3.4 Técnicas de extracción y selección de características.

El proceso de extracción de características documentado en (Bolón-Canedo, 2013), implica el mapeado de un espacio multidimensional a un espacio de menos dimensiones. Esto significa que el espacio de características original es transformado mediante la aplicación de una técnica de reducción de características, por ejemplo, utilizando la transformación lineal del Análisis de Componentes Principales PCA.

El proceso de extracción de características simplifica la cantidad de recursos necesarios para describir con precisión un amplio conjunto de datos. Lo que es necesario cuando se realiza un análisis de datos complejos, debido a que uno de los principales problemas del proceso de clasificación deriva del número de variables involucradas. Cuando se evalúa un considerable número de variables consecuentemente se requiere una gran cantidad de memoria y potencia de cálculo, por ello es importante controlar la cantidad de características que participan en el proceso de clasificación.

En el proceso de extracción de características se pueden extraer una inmensa cantidad de características. Es de vital importancia aplicar un proceso llamado selección de características , (Hota & Shrivas, 2014) lo define como un proceso de optimización que trata de encontrar el mejor subconjunto de características del conjunto fijo de las características según un determinado proceso objetivo y función original de los criterios de selección su objetivo es reducir el tamaño de los datos de entrada para facilitar el procesamiento y análisis de dicha información en pocas palabras de descartar aquellos datos no necesarios generando a si ahorro de tiempo y generación de resultados más óptimos.

2.3.1. Aplicación De Datasets En Sistemas De Detección De Intrusos.

De forma general, los datos son la materia prima bruta. En el momento que el usuario les atribuye algún significado especial pasan a convertirse en información. Cuando existen grandes cantidades de datos se hace muy difícil darle dicha atribución a ese inmenso conjunto de datos. Debido al crecimiento exponencial de los volúmenes de datos en las bases de datos organizacionales, dar sentido a la información valiosa se vuelve más y más difícil.

El descubrimiento de conocimiento en bases de datos KDD acrónimo en inglés Knowledge Discovery in Databases , nos permite identificar eficaz y coherentemente, patrones potencialmente útiles y previamente desconocidos en el proceso de KDD, con la aplicación de algoritmos específicos para la extracción de conocimiento deseable de conjuntos de datos para un fin en particular (Olson & Delen, 2008).

Tabla 2.2 *Datasets utilizados en procesos de simulación de Sistemas de Detección de Intrusiones.*

DATASET	**PATROCINADORES – MIEMBROS**
Dataset DARPA	• IST-LLMIT (Grupo de Tecnologías de Sistemas de Información- Laboratorio del Instituto de Tecnologías de Massachusetts). • DARPA ITO (Agenda de Proyectos de investigación Avanzada de Defensa-Oficina de Tecnología de la información). • AFRL/SNHS (Laboratorio de Investigación de las Fuerzas Aéreas).
Datasets USC/ISI ANT Programa PREDICT (Repositorio de Protección para la defensa de la infraestructura frente a	• ANT (Grupo de Investigación de Análisis de Tráfico de Red). • ISI (Instituto de Ciencia de la Información). • USC (Universidad del Sur de California) . • Departamento de Ciencias Computacionales de la

las amenazas informáticas) Proyecto LANDER	Universidad estatal de Colorado. • Departamento de Ingeniería Eléctrica de USC. • Servicios de Tecnologías de la Información USC.
Datasets CAIDA Asociación Cooperativa para el Análisis de Datos en Internet	Patrocinadores: ARIN (American Registry for Internet Member),CISCO , Endance Measurement Systems,U.S Departament of Homeland Security, NSF (National Science Fundation). Miembros: Digital Envoy, Intel, NIT(Nippon Telegraph and Telephone Corporation),Ripe NCC, University of California San Diego
Datasets CRAWDAD	• ACM SIGMOBILE • Intel Corporation • Fundación Nacional de Ciencias
Dataset DRDC Defense Research and Development Canada	• Seccion de operaciones de informacion de red (NIO) de la DRDC Ottawa, Canada. • Red de Establecimiento para la investigacion y Defensa (DREnet).
NIST SAMATE Reference Dataset Project NIST:National Institute for Standard and Tecnology SAMATE: Software Assurance Metrics and Tools Evaluation	Departamento de Estado EEUU
Virtual Dataset Repository	MERIT NETWORK INC. Programa PREDICT (Protected Repository for the Defense of Infraestructure again Cyber threats)

El dataset se utiliza para la evaluación de la eficiencia de los sistemas de dete

cción de intrusos en redes informáticas. Los criterios medibles son la probabilidad de detección y la probabilidad de falsas alarmas del respectivo sistema testeado. Aunque existe una amplia variedad de datasets los investigadores comúnmente se han decantado por el uso de DARPA NSL-KDD, por las ventajas que ofrece con respecto a otros dataset de su misma familia y de otras fuentes, en (Olson & Delen, 2008), (ENGEN, 2010) y (Franco, 2012) se hace un análisis en profundidad de las discrepancias encontradas en KDD cup '99, a partir de lo cual se aprecian las mejoras obtenidas en relación a la eliminación de datos redundantes e inconsistentes en NSL-KDD.

Adicional a lo anterior, la tabla siguiente 2.2 (Miguel & Hoz, 2012) muestra un listado de los dataset más destacados en procesos de simulación de sistemas de detección de intrusiones, esta información complementa el resumen de los conjuntos de datos más populares en el dominio de detección de intrusos que ha sido mostrada en (Wu & Banzhaf, 2010), (Zargari & Voorhis, 2012).

En esta investigación se ha decidido seleccionar el dataset NSL-KDD como insumo para las posteriores fases del proceso de simulación de detección de intrusiones. Dado que el LL-MIT ha demostrado considerables mejoras del dataset NSL-KDD respecto a sus antecesores y la comunidad investigadora mundial (en este ámbito de conocimiento) lo ha apropiado e implementando en sus investigaciones.

Tabla 2.3 *Estadística de artículos que utilizan el dataset NSL-KDD, Por base de dato.*

BASE DE DATO INDEXADA	**2011**	**2012**	**2013**	**2014**	**2015**
SCOPUS	10	16	16	32	12
SPRINGER	7	10	9	14	11
SCIENCEDIRECT	1	4	3	7	4
IEEEXPLORE DIGITAL LIBRARY	5	11	10	13	7

Evidencia de ello son las referencias que se hace a este dataset en artículos, paper de conferencias y paper in press, en diferentes bases de datos indexadas. A continuación, se muestra una tabla de los artículos por base de datos indexada, en los últimos cinco (5) años que manifiestan haber utilizado el dataset en el ámbito de la detección de intrusos (tabla 3.3).

2.3.2. *Dataset Darpa*

El Grupo de Tecnología de Sistemas de Información (IST), del Laboratorio Lincoln del Instituto Tecnológico de Massachusetts LL-MIT, con la cooperación de la Agencia de Proyectos de Investigación Avanzada de Defensa (DARPA ITO) y el Laboratorio de Investigación de las Fuerzas Aéreas (AFRL/SNHS), recopiló el primer dataset que contiene tráfico de red con una variada colección de conexiones. Los dataset publicados por LL-MIT en su web oficial, son los resultados de las evaluaciones en detección de intrusiones efectuada por DARPA en 1998 y 1999 (Singh, Singh, & BUIT, 2014).

También se encuentran experimentos dirigidos a escenarios específicos realizados en 2000. El LL-MIT distribuye libremente los dataset, la documentación, publicaciones, evaluaciones de resultados y herramientas de software relacionadas.

- **DATASET DARPA 1998:**

Según () ,el dataset DARPA 1998 contiene un conjunto de ataques realistas, integrados a un conjunto de conexiones normales, lo cual suministra el insumo de datos que permite evaluar las falsas alarmas y las tasas de detección de IDS; para construir este dataset se efectuaron dos evaluaciones: una off-line y otra en tiempo real.

La primera consta de tráfico de red y logs de auditoría recogidos en una red de simulación, para la segunda se insertaron sistemas de detección de intrusión en el banco de

pruebas de la red AFRL con la intensión de identificar sesiones de ataque en medio de actividades normales, en tiempo real.

Según (Lippmann et al., 1998), a partir de todos los datos recolectados se organizaron distintos subconjuntos de datos que componen el dataset DARPA 1998, tales como: datos de ejemplo, cuatro horas de subconjuntos de datos de entrenamiento, datos de entrenamiento (contienen siete semanas de ataques basados en red en medio de datos en segundo plano, normales) y datos de test (contiene dos semanas de ataques basados en red en medio de actividad normal en segundo plano)

- **DATASET DARPA 1999:**

El dataset DARPA 1999, al igual que su predecesor, está constituido por una evaluación off-line y una evaluación en tiempo real, basándose en los mismos principios que en el conjunto de datos del año anterior e incluyendo adicionalmente las siguientes características: ataques y tráfico desde ordenadores que ejecutan Windows NT, ataques en la red interna, archivos de sistema dump que proporcionan importantes componentes desde sistema de ficheros de cinco víctimas cada noche, incluyendo logs de auditoría de Windows NT y archivos de sniffing que proporcionan datos de sniffing de la red interna(Tavallaee & Bagheri, 2009).

Quedando el dataset DARPA 1999 constituido por: datos de entrenamiento (tres semanas de ataques, teniendo en cuenta que la primera y la tercera semana no contienen ataques, la segunda semana contiene un subconjunto selecto de ataques que van desde los ataques de 1998 a otros ataques nuevos), datos de test (dos semanas de ataques basados en red en medio de actividad normal en segundo plano).

- **DATASET DARPA NSL-KDD:**

El dataset NSL-KDD es una colección de datos construido con el objeto de solventar los problemas que presenta el conjunto KDD'99 (Singh et al., 2014) (MIT Lincoln Laboratory), pese a no ser una representación perfecta de los datos reales, debido a que no contiene conjuntos de datos públicos de los IDS; sin embargo, demuestra mucha utilidad al ser aplicado como un conjunto de datos de referencia eficaz para ayudar a los investigadores en el proceso de comparación de diferentes métodos de detección de intrusos.

El número de registros que contiene el dataset NSL-KDD es razonable, lo cual se constituye en una ventaja a la hora de realizar los experimentos con la colección de datos completa, para efectos de tiempo de procesamiento de la información, sin necesidad de elegir al azar a una pequeña porción de los datos, lo que consecuentemente conlleva a que los resultados de la evaluación de los trabajos de investigación lleguen a ser consistentes y comparables.

En (MIT Lincoln Laboratory) se encuentran los archivos de datos del NSL_KDD tanto en formato .txt como en formato .arff, cuya descripción se aprecia en la Tabla 2.4 .

Tabla 2.4 *Descripción de los tipos de KDD-Fuente:(www.darpa.mil)*

ARCHIVO	**DESCRIPCIÓN**
KDDTrain.arff	El conjunto de datos completo para el entrenamiento (Train NSL-KDD), con etiquetas binarias y en formato ARFF.
KDDTrain.txt	El conjunto de datos completo para el entrenamiento (train NSL-KDD), incluyendo etiquetas de tipos de ataques y el nivel de dificultad, en formato CSV.
KDDTrain_20Percent.arff	Un subconjunto del 20% del archivo KDDTRAIN.arff
KDDTrain_20Percent.txt	Un subconjunto del 20% del archivo KDDTrain.txt
KDDTest.arff	El conjunto de datos completo para el test con etiquetas

	binarias y en formato ARFF.
KDDTest.txt	El conjunto de datos completo para el test, incluyendo etiquetas de tipos de ataques y el nivel de dificultad, en formato CSV.
KDDTest21.arff	Un subconjunto del KDDTEST.arff el cual no contiene registros con el nivel de dificultad 21 de un total de 21.
KDDTest21.txt	Un subconjunto del archivo KDDTest.txt el cual no contiene registros con el nivel de dificultad 21 de un total de 21.

Capítulo 3

En este capítulo se abordan los ejes temáticos que fundamentan la investigación siguiendo un orden coherente con la investigación se definirá la conceptualización teórica de la

selección de características, seguidamente se abordaran las diferentes clasificaciones que existen y se realizara una introducción con relación a los conceptos de cada una de ellas con sus respectivas descripción en los diferentes tipos de técnicas de selección de características relacionadas con las clasificación, se representara de forma gráfica las ventajas y desventajas de los diferentes tipos de técnicas según su clasificación.

3.1 Selección de características

Según (De la Hoz Franco, De la Hoz Correa, Ortiz Garcia, Ortega Lopera, & Martinez Alvarez, 2014). La selección de característica también denominados atributos, componentes, variables, columnas, coordenadas o dimensiones, es un término usado habitualmente en minería de datos para describir las herramientas y las técnicas disponibles para reducir las entradas de los datos a un tamaño apropiado para su procesamiento y análisis. Se debe seleccionar o descartar activamente los atributos en función de su utilidad para el análisis.

La capacidad de aplicar la selección de características es esencial para un análisis eficiente, ya que los conjuntos de datos suelen contener mucha más información de la necesaria para la generación del modelo, ocasionando degradar la calidad de los patrones a detectar. Tanto las variables ruidosas, como las redundantes o correlacionadas y las variables irrelevantes, dificultan la detección de patrones significativos a partir de los datos.

Todo proceso de selección de atributos tiene un punto de partida, que puede ser el conjunto completo de atributos, el conjunto vacío o cualquier estado intermedio. Tras evaluar el primer subconjunto, se examinarán otros subconjuntos según una dirección de búsqueda, hacia adelante, hacia atrás, aleatoria o cualquier variación o mezcla de las anteriores.

El proceso terminará cuando se recorra todo el espacio o cuando se cumpla una condición de parada, según la estrategia de búsqueda seguida. Existen otros métodos de selección de atributos que se basan más en la transformación de los valores de entrada que en técnicas optimizadas de búsqueda, aportando información de cuanto de relevante es cada variable en su conjunto, pudiendo descartar aquellas que sean irrelevantes o que estén por debajo de un cierto umbral de relevancia.

3.1.1 Clasificación De Selección De Características

La selección de características en el análisis predictivo se refiere al proceso de identificación de las pocas variables más importantes o atributos que son esenciales en un modelo para una predicción precisa (Vijay & Bala, 2015). Por otra parte, se describen dos tipos de procesos de selección de características: Filter Type y Wrapper Type. La técnica por filtrado trabaja seleccionando sólo aquellos atributos que se encuentran entre la parte superior en el cumplimiento de ciertos criterios que figuren (Blum,1997; Yu, 2003). Según (Kohavi, 1997) la técnica de Envoltura (Wrapper Type) trabaja seleccionando de forma iterativamente, a través de un bucle de realimentación, sólo para aquellos atributos que mejoran el rendimiento de un algoritmo.

Según (Bolon Canedo, Sánchez Maroño, & Alonso Betanzos, 2013) realizando una revisión sobre los métodos de selección de función sobre datos sintéticos: Con respecto a la relación entre un algoritmo de selección de función y el aprendizaje inductivo método utilizado para inferir un modelo, tres enfoques principales se pueden distinguir:

Filtros:

Que dependen de las características generales de los datos de formación y llevan a cabo la función proceso de selección como una etapa de pre - procesamiento con independencia del algoritmo de inducción.

Envolturas:

Que implican la optimización de un predictor como parte del proceso de selección.

Métodos implícitos:

Son aquellos que se caracterizan por que realizan la función de selección en el proceso de formación y son generalmente específica para máquinas de aprendizaje dadas. Dentro de las técnicas de selección de características por filtrado se logra encontrar que existen diferentes métodos según (Bolon Canedo, Sánchez Maroño, & Alonso Betanzos, 2013) tales como se muestra en la figura 3.1.

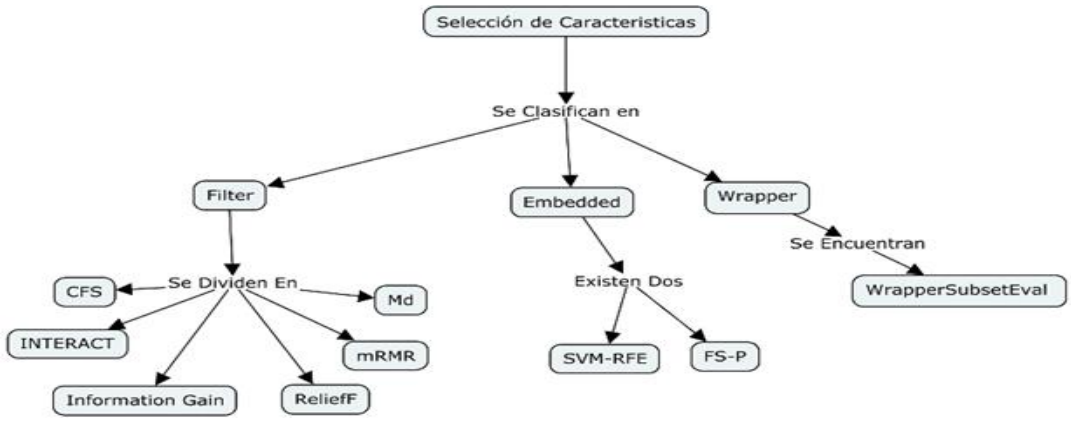

Figura 3.1 Clasificación de Selección de Características.

Fuente:(propia)

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 (Kaur, Kumar, & Kumar, 2015).

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 (Kaur, Kumar, & Kumar, 2015).

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.	CFS Interact ReliefF Information Gain

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

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*** (Bolón-Canedo, Sánchez-Maroño, & Alonso-Betanzos, 2012), describe las ventajas y desventajas y proporciona ejemplos, de cada una de las categorías de métodos de selección de características.

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

$$E(S) = \int_{\vec{S}} p(\vec{S}) \left(1 - \max_i \left(p(c_i|\vec{S})\right)\right) d\vec{S}, \quad (3.1)$$

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 (Hellman & Raviv, 1970), es:

$$E(\vec{S}) \leq \frac{H(C|\vec{S})}{2} \quad (3.2)$$

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

$$E(\vec{S};C) = H(C) - H(C|\vec{S}) \quad (3.3)$$

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 (Fano, 1961), 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*" (Duin, 2000).

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.

3.2 Descripción de las técnicas de selección utilizadas en la investigación de características

3.2.1. Métodos de filtrado:

En este apartado se propone un algoritmo para generar y refinar reglas de asociación que serán utilizadas para construir un clasificador para detección de estados emocionales. Los clasificadores suelen degradar su comportamiento ante atributos irrelevantes y/o redundantes. La selección de características es un proceso que consiste en seleccionar un subconjunto óptimo de características de una base de datos para reducir su dimensionalidad, eliminar ruido y mejorar el desempeño de un algoritmo de aprendizaje: velocidad de aprendizaje, precisión de la predicción y comprensibilidad de los resultados producidos.

El subconjunto óptimo de características está compuesto entonces por las características fuertemente relevantes y las débilmente relevantes pero no redundantes. Lo cual encontrar el subconjunto óptimo de características requiere una búsqueda en el espacio de subconjuntos posibles de características del conjunto de datos, que es un problema no determinístico polinomial complejo (A.L. Blum and P. Langley, 1992).

Básicamente un proceso de selección de características engloba una fase búsqueda y una fase de evaluación del subconjunto resultado de la búsqueda. Durante la fase de búsqueda se producen subconjuntos de características que son candidatos a ser evaluados. Existen diferentes tipos de búsquedas, pero podemos distinguir tres grandes tipos: la búsqueda completa garantiza

el hallazgo del subconjunto óptimo, sin tener la necesidad de realizar una búsqueda de todos los posibles subconjuntos (2*n) del total de n características, que es una búsqueda exhaustiva.

La búsqueda secuencial genera subconjuntos de manera directa, comienza con un subconjunto vacío, para luego agregarle características relevantes de manera progresiva (selección secuencial hacia adelante) o viceversa, comenzar con todo el conjunto y eliminar características irrelevantes de manera progresiva (selección secuencial hacia atrás) y por último la búsqueda aleatoria genera subconjuntos de manera aleatoria, luego aumenta o disminuye características también aleatoriamente para generar el siguiente subconjunto que sería evaluado.

Una vez finalizado el proceso de búsqueda se obtienen subconjuntos de datos que deben ser evaluados. El proceso de evaluación, consiste en medir que tan óptimo es el subconjunto generado para los fines de un problema de aprendizaje, que en este trabajo es de clasificación. Dicha clasificación divide las funciones de evaluación en dos categorías: *"Filtro"* y *"Wrapper"* (envolvente).

La diferencia entre una función de evaluación tipo *"Filtro"* o tipo *"Wrapper"* radica en que, en la primera categoría se incluyen los algoritmos en los que la selección de atributos se realiza como un pre-proceso a la fase clasificación y por tanto de manera independiente, por lo que puede entenderse como un filtrado de los atributos irrelevantes y redundantes. Por otro lado, en los métodos de tipo wrapper, la selección de atributos y los algoritmos de aprendizaje no son elementos independientes, ya que se utiliza el comportamiento de un algoritmo de clasificación como criterio de evaluación de los atributos. El modelo wrapper escoge los atributos que demuestran mejor clasificación, ayudando a mejorar el comportamiento del algoritmo de aprendizaje.

Además, serán descritos de manera general, los algoritmos de selección de características que se basan en medidas, para valorar la relevancia de una característica. A continuación, algunos de los más comúnmente utilizados.

- **Ganancia de información o información mutua:**

De la teoría de la información, se usa la cantidad de información que aporta una característica sobre la clase a predecir, para valorar la relevancia de dicha característica. Quinlan utilizaba la información mutua para elegir las características que dividirán nodos en generación de árboles (John R. Quinlan, 1986).

- **Gain Ratio**

La medida anterior de ganancia de información favorece a las características con muchos valores. Puede ocurrir que ésta sobre-estimación no sea un comportamiento deseable y para evitarlo se puede usar como medida el ratio entre la ganancia de información y la entropía de la característica. Esta medida fue usada por Quinlan para algoritmo como C4.5 formula *(F1)* (John R. Quinlan, C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993).

$$\text{Gain Ratio} = \frac{I(F;C)}{H(F)} \qquad \textbf{\textit{(3.4)}}$$

Según (Quinlan, 1993) Este método es muy utilizado y cuenta con popularidad en tener buenos resultados al clasificar, además se caracteriza por tener una particularidad donde una modificación de la ganancia de información que reduce su sesgo, la relación de ganancia toma el número y el tamaño de las ramas en cuenta a elegir un atributo su función se define como:

(1)

$$gan\ ratio(x) = gain(x)/sp(x) \quad (3.5)$$

Esta ecuación expresa la proporción de información generada por la división candidata que es útil. Este criterio selecciona el rasgo que maximice la razón (5.3) de donde

$$sp(x) = -\sum_{i=1}^{n} \frac{|Ti|}{|T|}.log\left(\frac{Ti}{T}\right) \quad (3.6)$$

$$gain(x) = i(T) - \sum_{i=1}^{n} \frac{|Ti|}{|T|}.i(Ti) \quad (3.7)$$

Donde la ecuación (5.5) favorece los atributos con muchos valores (generando un elevado número de divisiones candidatas), Gain ratio utiliza el tamaño de la división para normalizar la ganancia de información empleando (5.4).

- **Chi-Square**

Es una prueba estadística utilizada para comparar los datos observados con los datos que esperaríamos obtener de acuerdo con una hipótesis específica. Chi-Square siempre es la prueba de lo que los científicos llaman la hipótesis nula, lo que indica que no hay diferencia significativa entre el resultado esperado y el observado. La fórmula para calcular el Chi-Square es la que podemos apreciar en la formula *(F2)*:

$$Chi\ Square = (o-e)^2/e \quad \textbf{(3.8)}$$

Es decir, Chi-Square es la suma de las diferencias al cuadrado entre observada (O) y los datos esperados (e) (o la desviación, d), dividido por los datos esperados en todas las categorías posibles. Los datos utilizados en el cálculo de una estadística de Chi-Square deben ser al azar, crudo, mutuamente excluyentes, elaborado a partir de variables independientes y se extrae de una muestra lo suficientemente grande.

Hay básicamente dos tipos de variables aleatorias y con ellos se obtienen dos tipos de datos: numéricos y categóricos. Una estadística de Chi-Square se utiliza para investigar si las distribuciones de las variables categóricas se diferencian el uno del otro. Básicamente los datos de rendimiento variable categórico en las categorías y las variables numéricas producen datos en forma numérica (Sokolove, P. G., & Bushell, W. N, 1978). Chi-Square en pruebas nos permiten comparar las frecuencias observadas y esperadas de manera objetiva, ya que no siempre es posible saber con sólo mirar a ellos si son "lo suficientemente diferentes" para ser considerado estadísticamente significativo. La significación estadística en este caso implica que las diferencias no se deben a la casualidad, sino que pueden ser indicativos de otros procesos en el trabajo (Lancaster, H. O., & Seneta, E., 2005).

En (Namik & Othman, 2011) se utilizan reglas de asociación y Chi-square. Téngase presente que las reglas de asociación son una técnica popular para producir calidad en las detecciones basadas en mal uso (misused-based), sin embargo, la debilidad de las reglas de asociación radica en el hecho de que a menudo se producen miles de normas lo que reduce

el rendimiento de los IDS. (Namik & Othman, 2011) muestra una aplicación de post-minera para reducir el número de reglas y en consecuencia mejorando la calidad para producir firmas.

El experimento se llevó a cabo utilizando dos conjunto de datos recogidos de KDD Cup 99. Cada dataset se dividió en 4 grupos de datos en función del tipo de ataque (PROB, UR2, R2L y DOS). A su vez, cada partición utilizó Chi-Square como técnica de computación. La calidad de las normas se mide con base en el valor Chi-Square, que se calcula de acuerdo con al soporte, la confianza y la elevación de cada regla de asociación. Los resultados del experimento lograron reducir las reglas hasta el 98% permaneciendo la calidad de las normas.

$$X^2 = \sum \frac{(0 - E)^2}{E} \qquad (3.9)$$

Kumar (Muraleedharan, Parmar, & Kumar, 2010) presenta un sistema basado en el flujo de datos IP en una red computacional, para detectar actividad anómala utilizando chi-square, este sistema ofrece una solución para identificar actividades anómalas como ataques de exploración (scan) y de inundación (flood) por medio de análisis del comportamiento automático del tráfico de la red y también dando información detallada del atacante, la víctima, el tipo y momento del ataque que se puede utilizar para la defensa correspondiente.

En (Namik & Othman, 2011)se utilizan reglas de asociación y Chi-square. Téngase presente que las reglas de asociación son una técnica popular para producir calidad en las detecciones basadas en mal uso (misused-based), sin embargo, la debilidad de las reglas de asociación radica en el hecho de que a menudo se producen miles de normas lo que reduce el rendimiento de los IDS. (Namik & Othman, 2011) muestra una aplicación de post-minera para reducir el número de reglas y en consecuencia mejorando la calidad para producir firmas.

En (Oshima, Nakashima, & Nishikido, 2009) se realiza un extracción de características de anomalías de acceso en paquetes ip utilizando el método Chi-square, en el experimento se

verifico que un ataque Dos de la misma dirección ip fue detectado por el método, este análisis se centró en la dirección ip de origen para detectar la fuente del ataque de denegación de servicio, además se realizó un pre experimento basado en el escaneo de puertos en donde el método chi square pudo detectar ataques de escaneo de puertos.

- **Relief**

El algoritmo de Relief es presentado por Larry A Rendell (Kira, K. and L. A. Rendell, 1992), el cual se limita a los problemas de clasificación con dos clases. Se discute su extensión Relief (Kononenko, I, 1994), que puede hacer frente a los problemas multiclase. El algoritmo es capaz de tratar con datos incompletos y ruidosos.

Una idea clave del algoritmo Relief (Kira, 1992), dado en la Tabla 3.2, es que es función es estimar la calidad de los atributos de acuerdo con lo apropiado de sus valores, y estos se distinguen entre instancias que están cerca el uno al otro. Para ese propósito, dada una instancia seleccionada al azar Ri (línea 3), Relief busca por su dos vecinos más cercanos: uno de la misma clase llamada (Nearesthit H), y otra clase diferente llamada (Nearest miss M) (línea 4).

Tabla 3.2. *Pseudocódigo del algoritmo básico relief*

Algorithm Relief
Input: for each training instance a vector of attribute values and the class value *Output: the vector W of estimations of the qualities of attributes* *1. set all weights W[A] := 0.0;* *2. for i := 1 to m do begin* *3. randomly select an instance Ri ;* *4. find nearest hit H and nearest miss M;* *5. for A := 1 to a do*

6.	*W[A] := W[A]−diff(A,Ri ,H)/m+diff(A,Ri ,M)/m;*
7. end;	

Fuente: (Kira, K. and L. A. Rendell, 1992)

Se actualiza la estimación de calidad de W [A] para todos los atributos Una función de sus valores de Ri, M y H (líneas 5 y 6). Si las instancias Ri y H tienen diferentes valores de la A, a continuación el atributo A separa dos casos con la misma clase que no es deseable por lo que disminuimos la estimación de calidad de W [A] sobre él. Por otro lado, si las instancias Ri y M tienen diferentes valores del atributo A, Un atributo separa dos instancias con diferentes valores de la clase que es deseable por lo que aumentan la estimación de calidad de W [A]. Todo el proceso es repetido para m veces, donde m es un parámetro definido por el usuario.

En este enfoque (John, Kohavi, & Pfleger, 1994), 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.

De una manera empírica, mediante PCA se han logrado reducir la dimensionalidad en un amplio espectro de problemas de aprendizaje. En (Blum & Langley, 1997) 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***) (Comon, 1994), relacionado con el anterior, incorpora ideas similares, pero insistiendo en la independencia de las nuevas características en lugar de su ortogonalidad. En (De la Hoz, Ortiz, Ortega, & De la Hoz, 2013) 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.

- **Information gain:**

Es uno de los métodos de evaluación atributos más comunes. Este filtro univariado proporciona una clasificación ordenada de todas las características, y luego un umbral es requerido. En este trabajo el umbral se establecerá seleccionando las características que obtienen un valor de ganancia de información positiva.

Esta técnica de selección de características basadas en filtros (Hota & Shrivas, 2014) define info.gain como una técnica que mide las características de los datos a través de su ranking, basada en su fórmula (4.1) . El atributo con la mayor ganancia de información se elige como el atributo de división para el nodo N. Este atributo minimiza la información necesaria para clasificar las duplas en la partición resultante y refleja la menor aleatoriedad o impureza en estas particiones.

$$IG(D, X_3) = entropy(D) - \sum_v \frac{|D_v| entropy(D_v)}{|D|} \qquad (3.10)$$

- **One R:**

Este método basado y caracterizado por basarse en la tasa de error de las reglas generadas a partir de un conjunto de atributos a diferencia de otros algoritmos, la función solo contiene un

atributo se induce probando sobre el conjunto de entrenamiento todas las combinaciones de atributos y valore quedándose con la regla de menos errores, funciona como un clasificador de forma muy rápida; además de todo sus resultados son muy bueno en comparación con algoritmos muchos más complejos.

De igual manera este algoritmo también conocido en muchas revisiones documentales que se hicieron como 1R, propuesto por holte en 1993, es un clasificador es un clasificador muy sencillo, que únicamente utiliza un atributo para la clasificación. A pesar de que el autor lo cataloga como "Program 1R is ordinary in most respects." sus resultados pueden ser muy buenos en comparación con algoritmos mucho más complejos y su rendimiento promedio está por debajo de los de C4.5 en solo 5,7 puntos porcentuales de aciertos de clasificación según los estudios realizados por el autor del algoritmo (Holte, 1993).

La implementación del algoritmo 1R se realizó utilizando el pseudocódigo mostrado en la figura #3.2 Esta función solo permite trabajar con tablas que tengan atributos nominales y en las que no debe haber atributos con valores desconocidos para obtener el resultado deseado.

1r (Ejemplo){

```
Para cada atributo (A)
    Para cada tributo del (Ai)
        Contar el numero de apariciones de cada clase con Ai
        Obtener la clase mas frecuente (Cj)
        Crear una regla de tipo Ai->Cj
    Calcular el error de las reglas del atributo A
Escoger las reglas con menor error}
```

Figura #3.2 :“Pseudo Codigo del algoritmo 1R”.

Fuente: (Robles Aranda & Sotolongo, 2013)

Según (Robles Aranda & Sotolongo, 2013)La función toma como entrada el nombre de la tabla y la clase sobre la cual se va a realizar el análisis y devuelve como resultado un conjunto de reglas para los atributos con la menor cantidad de errores.

Capítulo 4

En este capítulo se explicara y se describe el proceso del modelo funcional planteado en esta investigación, además se detallan cada uno de los procesos que se encuentran por en cada fase, para este modelo encontramos tres fases, seguidamente encontraremos una descripción de cada una de las fases del modelo , plasmando una estructura funcional en donde se describen los procesos usados para el pre-procesamiento y normalización del Dataset KDD-train que pertenecen a la ***fase#1***, a su vez se describen los procesos de entrenamiento y clasificación que serían pertenecientes la ***fase #2***, y por último se evalúan las métricas de calidad de la propuesta que se encuentran inmersa en la ***fase#3*** del modelo.

4.1 Descripción de la propuesta

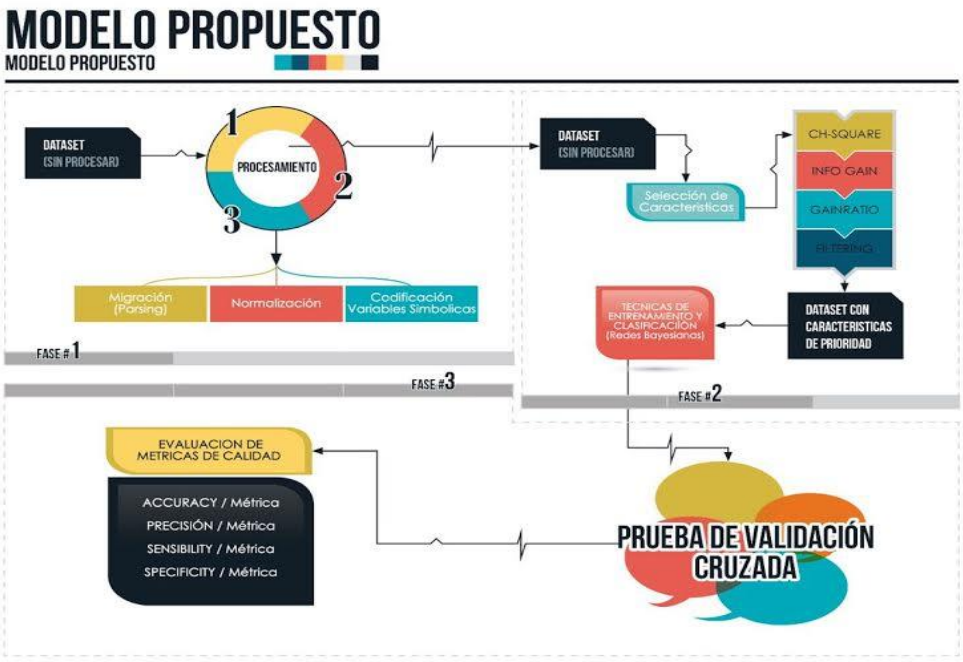

Figura 4.1 Modelo Propuesto.

Fuente:(Propia).

El modelo comprende tres fases: ***Fase#1*** (*Entrenamiento*), ***Fase#2*** (*Clasificación*) y ***Fase#3*** (*Cálculo de métricas de desempeño*). Para su aplicación se implementaron varios escenarios de simulación variando la cantidad de características a evaluar en las dos primeras fases, para ello se priorizó la escogencia de las características mediante los métodos de selección de características Chi square, Information gain , gainratio y Relief.

En la fase de clasificación se aplica el metodo de validacion cruzada con diez (10) pliegues, el cual representa el flujo de datos a clasificar, diferente del conjunto de datos de entrenamiento, se reducen las características usando la tasa discriminante de Fisher generada a partir del nuevo dataset, teniendo en cuenta la misma cantidad de características seleccionadas en la fase de entrenamiento y se procede por último a clasificar los datos, generando una estructura de datos que contiene tanto el etiquetado de la nueva data como el etiquetado predictivo a partir del cálculo de las BMUs basado en el mapa creado en la fase de entrenamiento.

En la fase final se calculan las métricas de desempeño para ello se recorre la estructura de datos generada en la fase anterior, calculando falsos positivos, verdaderos positivos, falsos negativos y verdaderos negativos, los cuales permiten determinar las métricas de sensibilidad, especificidad, exactitud y precisión que van a indicar la eficiencia del modelo planteado. Una descripción gráfica de lo anteriormente expuesto se aprecia en la **figura 4.1**

4.2 Descripción de la Fases

- **Fase #1**

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 (Bhuyan, Bhattachayya, & Kalita, 2013) (Theodoridis & Koutroumbas, Pattern Recognition, 4th Edition, 2009).

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:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \qquad (4.1)$$

Teniendo en cuenta lo anterior 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.

En la propuesta descrita, las 41 características del conjunto de datos NSL-KDD, son utilizadas en el algoritmo BAYES-NET para el cálculo de las distancias euclidianas, por tanto la escala de estas variables es muy importante para determinar la organización topológica del mapa.

Si el rango de valores de una variable es mucho más grande que el de las otras, ésta probablemente dominará la organización del mapa. La normalización de datos no permite que ninguna característica contribuya más que otra a la medida de la distancia. En (Vesanto, Himberg, Alhoniemi, & Parhankangas, 2000) se presentan seis implementaciones de métodos de normalización: var, range, log, logistic, histD e histC.

Var, normaliza la varianza de la variable a la unidad y su media a cero. Esto es una transformación lineal simple, como se indica en (4.2).

$$x\hat{}=(x-\bar{x})/\sigma \qquad (4.2)$$

Donde $\bar{x}$ y σ son, respectivamente, la media y la desviación estándar de la variable x. Esto es equivalente a expresar la variable x como la distancia entre el número de desviaciones estándar y su media.

Range, escala los valores de la variable entre [0, 1] con una transformación lineal simple, como se indica en (2.6). Los parámetros de la transformación son: los valores mínimos y el rango max(x)-min(x) de la variable. Si la transformación es aplicada a nuevos datos con valores por fuera del rango mínimo y máximo, los valores de la transformación estarán también por fuera del rango [0, 1].

$$x'=(x-\min(x))/(\max(x)-\min(x)) \qquad (4.3)$$

Log, es una transformación logarítmica útil si los valores de la variable se distribuyen de forma exponencial con valores demasiado pequeños y un número menor de valores grandes. Esta transformación es una buena forma de conseguir mayor resolución en la parte baja del vector de componentes.

Lo que en realidad se hace es una transformación no lineal, ver ecuación (4.4), donde ln es el logaritmo natural, generando valores no-negativos. Se debe tener especial cuidado cuando la transformación es aplicada a un nuevo conjunto de datos, con valores por debajo de min(x)-1, dado que los resultados serán números complejos. Por lo tanto, si se conocen los valores previamente, es conveniente asignarles manualmente el valor mínimo min(x).

$$x'=\ln(x-\min(x)+1) \qquad (4.4)$$

Logistic, también denominada normalización softmax. Este método de normalización se asegura de que todos los valores, desde menos infinito hasta más infinito se encuentren dentro del rango [0, 1]. La transformación es más o menos lineal en la mitad del rango (alrededor del valor medio), y tiene una linealidad suavizada en ambos extremos, lo cual asegura que todos los valores estén dentro del rango.

El dato primero se escala como una normalización de varianza, ver ecuación (4.2), luego se aplica la función logistic, ver ecuación (4.5). Los parámetros de la transformación son el valor medio $\bar{x}$ y la desviación estándar σ de los valores originales, tal como en el método de normalización var.

x'=1/((1+e^(-xˆ))) (4.5)

histD, es una ecualización de histograma discreta. Que ordena los valores y luego sustituye cada uno por su número ordinal. Por último, escala linealmente los valores de modo que estén entre [0,1]. Útil tanto para variables discretas como continuas, sin embargo como los parámetros de transformación son todos valores únicos del conjunto de datos de inicialización, esto puede requerir el uso de una considerable cantidad de memoria.

Si la variable puede tomar unos pocos valores (20, por ejemplo) puede que sea mejor utilizar el método histC. El método histD no es exactamente reversible, si es aplicado a valores que no hagan parte del conjunto de valores originales.

histC es una ecualización de histograma continua. Realmente, es una transformación lineal parcial, la cual trata de ser como una ecualización de histograma. 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. Por ejemplo, valores en el contenedor número 3 son escalados entre [3,4]. Finalmente, todos los valores son linealmente escalados entre [0,1].

El número de contenedores es el redondeo de la raíz cuadrada del número de valores únicos en el conjunto de inicialización. La ecualización de histograma resultante no es tan buena como la que hace histD, pero el beneficio es que es exactamente reversible, incluso fuera del rango de valores originales.

En esta tesis se utilizó la implementación var, también denominada whitening o normalización a media cero y varianza unidad. La cual se ha seleccionado debido a que genera un mejor desempeño en las pruebas experimentales respecto a las otras implementaciones. En este trabajo, las variables continuas son normalizadas con media cero y varianza unitaria utilizando la ecuación (4.2).

Por otra parte, todas las variables se escalan en el intervalo [0.1]. Las características simbólicas (ya codificadas con valores binarios) y las binarias no son normalizadas. Una vez normalizados los datos, se efectuó un análisis comparativo de dos técnicas de selección de características, que se describen en detalle a continuación.

- **Fase # 2**

Luego de realizar el proceso que se explicó en la fase#1 en el que se obtienen como resultado sobre un conjunto de datos (Dataset KDD-NLS), depurado y listo para que se le apliquen las técnicas de selección de características que este caso son Chi square, Info.Gain, Gain Ratio, One R y Relief seguidamente como resultado de este proceso obtendremos un cojunto de datos (Dataset) priorizado después de aplicarles las diferentes técnicas de características., a continuación se realizara la descripción de una de las técnicas aplicadas a esta investigación.

- **Fase # 3**

En esta fase final del proceso de simulación, se calculan las métricas de calidad del modelo propuesto, teniendo como argumento de entrada los datos arrojados por la clasificación. Para ello fue necesario calcular la cantidad de: verdaderos positivos (TP), verdaderos negativos (TN), falsos positivos (FP) y falsos negativos (FN). A partir de lo anterior se calcularon las métricas (sensibilidad, especificidad, exactitud y precisión) con el propósito de conocer la capacidad de clasificación del modelo propuesto, y permitir validar su eficiencia

4.3 Métricas de desempeño

En esta investigación se usan métricas de desempeño estadísticas, para medir el comportamiento del IDS en relación al proceso de clasificación,en coherencia con lo planteado por (Andersen, Glasdam, & Larsen, 2016),(Eid, Hassanien, Kim, & Banerjee, 2013) y (Panda, Abraham, & Patra, 2010);tales métricas se definen a continuación.

- **SENSIBILIDAD**: Define sensibilidad como la capacidad que tiene un IDS para identificar resultados "*verdaderos positivos*":

$$Sensibilidad = \frac{VP}{VP + FN} \qquad (4.6)$$

- **ESPECIFICIDAD**: Define especificidad como la capacidad que tiene un IDS de medir la proporción de "*verdaderos negativos*" que se han identificado correctamente:

$$Especificidad = \frac{VN}{VN + FP} \qquad (4.7)$$

- **EXACTITUD**: (Andersen et al., 2016) define exactitud como el grado de cercanía de las mediciones de una cantidad (X) al valor de la magnitud real (Y); Es decir la proporción de resultados verdaderos (tanto verdaderos positivos como verdaderos negativos). Una exactitud del 100% significa que los valores medidos son exactamente los mismosque losvalores dados :

$$Exactitud = \frac{VP + VN}{VP + FP + FN + VN} \qquad (4.8)$$

- **PRECISIÓN**: Define la proporción de verdaderos positivos contra todos los resultados positivos:

$$Precisión = \frac{VP}{VP + FP} \quad (4.9)$$

Capítulo 5

5. Escenarios de experimentación.

En el presente capitulo se recrean y analizan distintos escenarios de simulación para la detección de intrusos en sistemas computacionales aplicando el modelo propuesto utilizando el dataset DARPA NSL-KDD, en el cual se detalla cada escenario aplicando variación de técnicas de selección de características ***(Chi Square, Info.Gain, Gain Ratio, Relieff, Simmetrical Uncert, One-R Y Filtered)*** . En dicho proceso se utiliza como técnica de entrenamiento las ***redes bayesianas*** y la técnica de ***validación cruzada 10 (diez) pliegues*** como método para sistema de simulación en laboratorio.

En la **sección 5.1** se utilizó validación cruzada con 10 pliegues, de los cuales 9 pliegues para entrenamiento y uno para las pruebas, con el objetivo de efectuar comparaciones entre los diferentes escenarios planteados en el modelo propuesto. Una vez implementadas las técnicas de selección, la técnica de entrenamiento y clasificación anteriormente mencionada se procedió a evaluar unas métricas de desempeño con el objetivo de identificar la eficiencia del modelo propuesto en el cual se enfocan los mejores resultados en las métricas de desempeño empleadas para valorar la eficiencia del modelo.

En la **sección 5.2** se realiza una consolidación de todos los resultados de simulación con el propósito de identificar el modelo más efectivo con respecto a los mejores resultados de las métricas evaluadas variando la técnica de selección de atributos.

5.1 Escenarios experimentales (conjunto de características seleccionadas, clasificando con redes bayesianas y aplicando validación cruzada).

Por ello, a continuación, se plantean los mismos escenarios de simulación anteriormente analizados, pero aplicando validación cruzada k-pliegues (k = 10). Ya que k = 10, contiene particiones del 90% de las muestras que fueron seleccionadas aleatoriamente para ajustar el modelo; el resto de las muestras (10%) se utilizó para la prueba.

Estos subconjuntos son diferentes y no comparten ninguna muestra. Este proceso se repitió para los 10 pliegues, asegurándose de que los datos de prueba nunca se hubiesen utilizado en 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 10.

Si bien es cierto que la exactitud es la proporción de resultados verdaderos (tanto verdaderos positivos, como verdaderos negativos). Esta métrica es por tanto, la más preponderante para el estudio de tráfico en redes computacionales y es en definitiva de alta relevancia para la toma de decisión de una prueba de laboratorio.

5.1.1 Escenario experimental 1: simulación clasificación REDES BAYESIANAS con validación cruzada

En este escenario se consideran las 41 características del dataset *KDD-Train al* 100% y realizando la clasificación utilizando la técnica de ***REDES BAYESIANAS***. Para la validación cruzada se aplicaron diez (10) pliegues y el resultado de las métricas se calculó a partir de la media de los resultados parciales

Tabla No. 5.1. *Resultado de prueba de simulación aplicando REDES BAYESIANAS con validación cruzada*

METODO		CARACT		PRECISION	EXACTITUD	SENSIBILIDAD	ESPECIFICIDAD
REDES		41		96,70%	96,56%	94,69%	98,93%

BAYESIANAS							

Como se observa en la **Tabla No. 5.1**. , la simulación aplicando la técnica de redes bayesianas plantea unos resultados interesantes en sus métricas, con unos índices de precisión de 96,70%, exactitud de 96,56%, sensibilidad en 94,69% y especificidad con 98,93%. Esto último significa que la clasificación con redes bayesianas, utilizando todas sus características, detecta el tráfico normal en un porcentaje de 98,93%.

5.1.2 escenario experimental 2: simulación chi square + REDES BAYESIANAS con validación cruzada

Tabla No 5.2. *Resultados de prueba de simulación aplicando CHI SQUARE+REDES BAYESIANAS con validación cruzada*

METODO	**CARACT.**	**PRECISION**	**EXACTITUD**	**SENSIBILIDAD**	**ESPECIFICIDAD**
CHI SQUARE	**5**	96,40 %	96,31%	97,89%	94,60%
	10	95,30 %	95,23%	97,92%	92,16%
	15	94,00 %	93,53%	99,14%	87,12%
	16	94,40 %	93,95%	99,03%	88,12%
	17	95,10 %	94,78%	99,14%	89,77%

18	96,30 %	96,11%	99,31%	92,46%
19	96,30 %	96,19%	99,24%	92,69%
20	95,90 %	95,71%	99,32%	91,58%
25	96,50 %	96,35%	99,10%	93,20%
30	96,70 %	96,55%	99,11%	93,61%
35	96,70 %	96,56%	99,12%	93,64%
41	96,70 %	96,56%	99,12%	93,64%

Se desarrolló una simulación tomando el dataset KDD-Train 100 % y aplicando la técnica de selección de característica CHI SQUARE. Con esta técnica se pudo identificar el orden de prioridad de las características del dataset , lo cual permitió variar el número de ellas generando pruebas con 5,10,15,16,17,18,19,20,25,30,35 y 41 características .

Teniendo en cuenta tabla 5.2 se genera el mejor resultado a 35 características con sensibilidad 99,12%, especificidad 93,64%, precisión 96,70% y exactitud de 96,54%.

5.1.3. escenario experimental 3: simulación info.gain + redes bayesianas con validación cruzada.

Se desarrolló una simulación tomando el dataset KDD-Train 100 % y aplicando la técnica de selección de característica INFO.GAIN. Con esta técnica se pudo identificar el orden de prioridad de las características del dataset , lo cual permitió variar el número de ellas generando pruebas con 5,10,15,16,17,18,19,20,25,30,35 y 41 características .

Tabla No 5.3. *Resultados de prueba de simulación aplicando INFO.GAIN +REDES BAYESIANAS con validación cruzada*

METODO	CARACT.	PRECISION	EXACTITUD	SENSIBILIDAD	ESPECIFICIDAD
INFO.GAIN	**5**	96,40%	96,31%	95,14%	97,65%
	10	94,10%	93,73%	98,53%	88,24%
	15	94,00%	93,53%	99,14%	87,12%
	16	94,40%	93,95%	99,03%	88,12%
	17	95,10%	94,78%	99,14%	89,77%
	18	96,30%	96,11%	99,31%	92,46%
	19	96,30%	96,19%	99,24%	92,69%
	20	95,90%	95,71%	99,32%	91,58%
	25	96,50%	96,35%	99,10%	93,20%
	30	96,70%	96,55%	99,11%	93,61%
	35	96,70%	96,56%	99,12%	93,64%
	42	96,70%	96,56%	99,12%	93,64%

Teniendo en cuenta la ***tabla 5.3*** se genera el mejor resultado a 35 características con sensibilidad 99,12%, especificidad 93,64%, precisión 96,70% y exactitud de 96,56%.

5.1.4. Escenario experimental 4: simulación gain ratio + REDES BAYESIANAS con validación cruzada

Se desarrolló una simulación tomando el dataset KDD-Train 100 % y aplicando la técnica de selección de característica GAIN RATIO. Con esta técnica se pudo identificar el orden de prioridad de las características del dataset , lo cual permitió variar el número de ellas generando pruebas con 5,10,15,16,17,18,19,20,25,30,35 y 41 características .

Teniendo en cuenta la ***tabla 5.4*** se genera el mejor resultado a 30 características con sensibilidad 99,14%, especificidad 94,06%, precisión 96,90% y exactitud de 96,77%.

Tabla No 5.4. *Resultados de prueba de simulación aplicando GAIN RATIO +REDES BAYESIANAS con validación cruzada*

METODO	CARACT.	PRECISION	EXACTITUD	SENSIBILIDAD	ESPECIFICIDAD
GAIN RATIO	**5**	87,70%	84,93%	99,06%	68,75%
	10	92,60%	92,55%	95,54%	89,13%
	15	94,40%	93,96%	98,95%	88,25%
	16	94,50%	94,18%	98,90%	88,77%
	17	94,50%	94,18%	98,79%	88,91%
	18	94,40%	94,07%	98,78%	88,67%
	19	94,40%	93,96%	98,95%	88,25%
	20	94,80%	94,42%	99,09%	89,07%
	25	95,70%	95,44%	99,20%	91,14%

	30	96,90%	96,77%	99,14%	94,06%
	35	96,70%	96,55%	99,12%	93,61%
	42	96,70%	96,56%	99,12%	93,64%

5.1.5. escenario experimental 5: simulación relieff+ REDES BAYESIANAS con validación cruzada

Tabla No 5.5. *Resultados de prueba de simulación aplicando RELIEFF + REDES BAYESIANAS con validación cruzada*

METODO	CARACT.	PRECISION	EXACTITUD	SENSIBILIDAD	ESPECIFICIDAD
RELIEFF	**5**	93,90%	93,38%	99,03%	89,49%
	10	93,90%	93,38%	99,03%	89,49%
	15	92,90%	92,19%	98,58%	87,94%
	16	92,50%	91,75%	98,56%	87,30%
	17	93,00%	92,21%	98,75%	87,88%
	18	92,90%	92,21%	98,73%	87,89%
	19	92,90%	92,22%	98,66%	87,95%
	20	93,10%	92,44%	98,37%	88,43%
	25	93,20%	92,60%	98,51%	88,59%
	30	93,80%	93,30%	98,40%	89,71%
	35	96,60%	96,54%	98,92%	94,67%
	42	96,70%	96,56%	98,93%	94,69%

Se desarrolló una simulación tomando el dataset KDD-Train 100 % y aplicando la técnica de selección de característica RELIEFF. Con esta técnica se pudo identificar el orden de prioridad de las características del dataset , lo cual permitió variar el número de ellas generando pruebas con 5,10,15,16,17,18,19,20,25,30,35 y 41 características .

Teniendo en cuenta la premisa anterior se genera el mejor resultado a 35 características con sensibilidad 98,92%, especificidad 94,67%, precisión 96,60% y exactitud de 96,54 % como se denota en ***tabla 5.5.***

5.1.6. Escenario experimental 6: simulación simmetrical uncert + REDES BAYESIANAS con validación cruzada

Se desarrolló una simulación tomando el dataset KDD-Train 100 % y aplicando la técnica de selección de característica SIMMETRICAL UNCERT. Con esta técnica se pudo identificar el orden de prioridad de las características del dataset , lo cual permitió variar el número de ellas generando pruebas con 5,10,15,16,17,18,19,20,25,30,35 y 41 características .

Tabla No 5.6. *Resultados de prueba de simulación aplicando SIMMETRICAL UNCERT + REDES BAYESIANAS con validación cruzada*

METODO	CARACT.	PRECISION	EXACTITUD	SENSIBILIDAD	ESPECIFICIDAD
SYMMETRICAL UNCERT	**5**	96,20%	96,17%	94,60%	97,61%
	10	92,60%	92,49%	94,58%	90,83%
	15	94,50%	94,13%	98,59%	90,92%
	16	94,40%	93,95%	98,76%	90,52%
	17	95,10%	94,78%	98,92%	91,74%
	18	95,10%	94,81%	98,87%	91,82%

19	96,30%	96,19%	99,07%	93,95%
20	96,20%	96,06%	99,07%	93,75%
25	96,50%	96,35%	98,91%	94,35%
30	96,70%	96,55%	98,92%	94,67%
35	96,70%	96,56%	98,93%	94,69%
42	96,70%	96,56%	98,93%	94,69%

Teniendo en cuenta la tabla 5.6 se genera el mejor resultado a 30 características con sensibilidad 98,92%, especificidad 94,67%, precisión 96,70% y exactitud de 96,55 % .

5.1.7. Escenario experimental 7: simulación one-r + redes bayesianas con validación cruzada

Se desarrolló una simulación tomando el dataset KDD-Train 100 % y aplicando la técnica de selección de característica ONE R. Con esta técnica se pudo identificar el orden de prioridad de las características del dataset , lo cual permitió variar el número de ellas generando pruebas con 5,10,15,16,17,18,19,20,25,30,35 y 41 características .

Teniendo en cuenta la tabla 5.7 se genera el mejor resultado a 30 características con sensibilidad 98,53%, especificidad 95,23%, precisión 96,80% y exactitud de 96,71 %.

Tabla No 5.7. Resultados de prueba de simulación aplicando SIMMETRICAL UNCERT + REDES BAYESIANAS con validación cruzada

METODO	CARACT.	PRECISION	EXACTITUD	SENSIBILIDAD	ESPECIFICIDAD
ONE R	**5**	96,20%	96,16%	94,32%	97,88%

10	95,30%	95,23%	97,48%	93,46%
15	94,10%	93,70%	98,68%	90,19%
16	94,40%	94,02%	98,86%	90,57%
17	94,80%	94,45%	99,11%	91,11%
18	96,50%	96,36%	99,00%	94,31%
19	96,40%	96,24%	99,10%	94,02%
20	96,40%	96,25%	98,99%	94,12%
25	96,70%	96,58%	98,52%	95,02%
30	96,80%	96,71%	98,53%	95,23%
35	96,80%	96,70%	98,52%	95,23%
42	96,80%	96,70%	98,52%	95,23%

5.1.8. Escenario experimental 8: simulación filtering + redes BAYESIANAS con validación cruzada

Se desarrolló una simulación tomando el dataset KDD-Train 100 % y aplicando la técnica de selección de característica FILTERING. Con esta técnica se pudo identificar el orden de prioridad de las características del dataset , lo cual permitió variar el número de ellas generando pruebas con 5,10,15,16,17,18,19,20,25,30,35 y 41 características .

Tabla No 5.8. *Resultados de prueba de simulación aplicando FILTERING + REDES BAYESIANAS con validación cruzada*

METODO	CARACT.	PRECISION	EXACTITUD	SENSIBILIDAD	ESPECIFICIDAD
FILTERING	**5**	96,40%	96,31%	97,89%	94,60%

10	94,10%	93,73%	98,53%	88,24%
15	94,00%	93,53%	99,14%	87,12%
16	94,40%	93,95%	99,03%	88,12%
17	95,10%	94,78%	99,14%	89,77%
18	96,30%	96,11%	99,31%	92,46%
19	96,30%	96,19%	99,24%	92,69%
20	95,90%	95,71%	99,32%	91,58%
25	96,50%	96,35%	99,10%	93,20%
30	96,70%	96,55%	99,11%	93,61%
35	96,70%	96,56%	99,12%	93,64%
42	96,70%	96,56%	99,12%	93,64%

Teniendo en cuenta la premisa anterior se genera el mejor resultado a 35 características con sensibilidad 99,12%, especificidad 93,64%, precisión 96,70% y exactitud de 96,56 % como se denota en tabla 5.8.

5.2 Consolidación de resultados experimentales.

Tabla No 5.9. *Consolidación de experimentos aplicando BAYES NET con VALIDACIÓN CRUZADA*

METODO	CARACT	PRECISION	EXACTITUD	SENSIBILIDAD	ESPECIFICIDAD
CHI SQUARE+ RED	35	96,70%	96,56%	99,12%	93,64%

BAYESIANA					
INFO.GAIN+RED BAYESIANA	**35**	96,70%	96,56%	99,12%	93,64%
FILTERING+ RED BAYESIANA	**35**	96,70%	96,56%	99,12%	93,64%
GAIN RATIO + RED BAYESIANA	**30**	**96,90%**	**96,77%**	**99,14%**	**94,06%**
S. UNCERT+ RED BAYESIANA	**30**	96,70%	96,55%	98,92%	94,67%
ONE R + RED BAYESIANA	30	96,80%	96,71%	98,53%	95,23%
RELIEFF+ RED BAYESIANA	**35**	96,60%	96,54%	98,92%	94,67%

En la ***tabla 5.9*** se comparan los mejores resultados obtenidos de los escenarios de simulación anteriormente expuestos en la ***sección 5.1***. A partir de ello se deduce que la propuesta GAIN RATIO + RED BAYESIANA utilizando 30 características genera los mejores resultados en las métricas de evaluación.

6. Conclusiones

En el presente capítulo se plasman las conclusiones a las cuales se ha llegado el desarrollo de la tesis de investigación, en el cual se describe como alcanzamos cada uno de nuestros objetivos tabla, se anuncian los resultados obtenidos, a su vez se plantean trabajos futuros para así seguir dándole continuidad a la línea de investigación objeto de estudio.

Los estudios orientados hacia los sistemas de detección de intrusos son de gran beneficio para garantizar la seguridad de los diferentes tipos de redes informáticas debido a los aportes que generan sus resultados con la aplicación de técnicas de selección de características y clasificación, los cuales son de gran importancia para la construcción de sistemas más eficientes.

Al momento de realizar los distintos escenarios de simulacion utilizando variacion en las tecnicas de selección ***(Chi Square, Info.Gain, Gain Ratio, Relieff, Simmetrical Uncert, One-R Y Filtered)*** *con la tecnica de entrenamiento y clasificacion* Redes Bayesiana (Bayes Net), se obtuvo que aplicando 30 (treinta) caracteristicas genera los mejores resultados aplicando 30 características generó los mejores resultados en las métricas teniendo en cuenta la precisión 96,90% en especificidad 94,06, sensibilidad de 99,14 % y 96,77% en exactitud, siendo así el modelo más eficiente para la detección de ataques de intrusos basados en anomalías.

En relación con lo anterior esta investigación tiene en cuenta métricas muy importantes las cuales generan conocimiento para tener en cuenta en futuras investigaciones.

7. Trabajos futuros

Se pretende trabajar a futuro desarrollando un modelo como el SOM actual y además identificar si podría mejorarse mediante hibridación con otras técnicas de entrenamiento, tales como el uso de Maquinas de Soporte Vectorial (SVM), arboles de decisión.

También se pretende realizar una arquitectura tecnológica, que permita capturar coleccionar e identificar el tráfico de red, con el fin de clasificar cuáles serían los ataques y el tráfico normal que se presentan en un campus universitario.

Referencias

De La Hoz, E., De la Hoz, E., Ortiz, A., Ortega, J., & Prieto, B. (21 de September de 2015). PCA filtering and probabilistic SOM for network intrusion detection. *Neurocomputing, 164,* 71-81. doi:10.1016/j.neucom.2014.09.083

Devijver, P., & Kittler, J. (1982). *Pattern Recognition: A Statistical Approach.* Londres: Prentice-Hall.

Shlens, J. (2009). A Tutorial on Principal Component Analysis. Center for Neural Science, NYU y Systems Neurology Laboratory, Salk Institute for Biological Studies La Jolla.

Alahakoon, D., Halgamuge, S., & Srinivasan, B. (1998). A structure adapting feature map for optimal cluster representation. *International Conference on Neural Information Processing ICONIP98*, 809-812.

Alhoniemi, E., Himberg, J., & Vesanto, J. (1999). Probabilistic measures for responses of self-organizing map units. *Proceedings of the International ICSC Congress on Computational Intelligence Methods and Applications (CIMA), 1*, 286-290.

Alvarez Illán, I. (Junio de 2009). Análisis en Componentes de Imágenes Funcionales para la Ayuda al Diagnóstico de la Enfermedad del Alzheimer. *Tesis Doctoral*. Granada.

Álvarez Illán, I., Manuel Górriz, J., Ramírez, J., Salas GonzáLez, D., López, M. M., Segovia, F., . . . Puntonet, C. (February de 2011). 18F-FDG PET imaging analysis for computer aided

Alzheimer's diagnosis. *Information Sciences, 181*(4), 903-916. doi:10.1016/j.ins.2010.10.027

Anderson, J. (1980). *Computer Security Threat Monitoring and Surveillance.* Fort Washington, Pennsylvania: James P. Anderson Company.

Axelsson, S. (2000). *Intrusion Detection Systems: A Taxonomy and Survey.* Technical Report 99-15, Dept. of Computer Engineering, Chalmers University of Technology, Goteborg, Sweden.

Bace, R. (2000). *An Introduction to Intrusion Detection and Assessment / for System and Network Security Management.* Obtenido de ICSA: http://www.iss.net/documents/whitepapers/intrusion.pdf

Ben-Hur, A., & Guyon, I. (2003). *Detecting stable clusters using principal component analysis.* (M. Brownstein, & A. Kohodursky, Edits.) Humana press.

Bhuyan, M., Bhattachayya, D., & Kalita, J. (2013). Network anomaly detection: methods , systems and tools. *IEEE Commun. Surv. Tutor*, 99.

Blackmore, J., & Miikkulainen, R. (1993). Incremental grid growing: Encoding high-dimensional structure into a two-dimensional feature map. *Proceedings of the International Conference on Neural Networks ICNN93, I*, 450-455.

Blum, A., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 245-271.

Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2012). A review of feature selection methods on synthetic data. *Knowledge and Information System*, 483-519.

Bouckaert, R. (2008). Practical bias variance decomposition. *Advances in Artificial Intelligence - LNCS., 5360*, 247-257.

Bouvrie, P., Angulo, J., & Dehesa, J. (1 de June de 2011). Entropy and complexity analysis of Dirac-delta-like quantum potentials. *Physica A: Statistical Mechanics and its Applications, 390*(11), 2215–2228. doi:doi:10.1016/j.physa.2011.02.020

Bouzida, Y., & Gombault, S. (2004). Eigenconnections to intrusion detection. *19th IFIP International Information Security Conference (SEC2004), IEEE, 147*, págs. 241–258. Toulouse, France. doi:10.1007/1-4020-8143-X_16

Bradley, P., & Fayyad, U. (1998). Refining initial points for K-Means clustering. *Proc. 15th International Conf. on Machine Learning* (págs. 91–99). San Francisco, CA: Morgan Kaufmann. Obtenido de citeseer.ist.psu.edu/bradley98refining.html

Breiman, L., Friedman, J., Stone, C., & Olshen, R. (1984). *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).

Brumlen, D., Wang, H., Newsome, J., & Song, D. (2006). Towards Automatic Generation of Vulnerability-based Signatures. *IEEE Symposium*, 1081-6011.

Buenabad, J., & Coria, J. (Junio de 2004). Tolerancia a fallas para sistemas de detección de intrusos de red. *Tesis de Maestría*. CINVESTAV-IPN.

California, U. O. (1999). *The UCI KDD Archive*. (University of California) Obtenido de http://kdd.ics.uci.edu/databases/kddcup99/task.html

Calvo, R. F. (9 de Septiembre de 2000). *ati*. Obtenido de http://www.ati.es/novatica/glosario/glosario_internet.txt

Cano, J., Herrera, F., & Lozano, M. (15 de May de 2005). Stratification for scaling up evolutionary prototype selection. *Pattern Recognition Letters, 26*(7), 953-963. doi:10.1016/j.patrec.2004.09.043

Carpenter, G., & Grossberg, S. (1988). The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network. *Computer, 21*(3), 77-88.

Chapman, D., & Zwicky, D. (1997). *Construya Firewalls para Internet.* Mexico: MacGraw-Hill.

Chauhan, H., & Chauhan, A. (2013). Implementation of decision tree algorithm c4.5. *International Journal of Scientific and Research Publications, 3*(10).

Cheng, S.-S., Fu, H.-C., & Wang, H.-M. (2009). Model-Based Clustering by Probabilistic Self-Organizing Maps. *IEEE TRANSACTIONS ON NEURAL NETWORKS, 20*(5), 805-826.

Chet, H., & Duren, M. (1998). Detecting Subtle System Changes Using Digital Signatures. *Information Technology Conference, IEEE*, 125-128.

Choi, S.-S., Cha, S.-H., & Tappert, C. (2010). A survey of binary similarity and distance measures. *Systemics, Cybernetics And Informatics, 8*(1), 43-48.

Comon, P. (1994). Independent component analysis, a new concept? *Signal Process, 36*(3), 287-314.

Computer Security Resource Center. (15 de Abril de 1980). *Computer Security Threat Monitoring and Surveillance.* Obtenido de http://csrc.nist.gov/publications/history/ande80.pdf

ComputerWire. (2002). *DDoS Really, Really Tested UltraDNS. Informe técnico*. Obtenido de http://www.theregister.co.uk/2002/12/14/ddos_attack_really_really_tested/ attack really really tested

Cost, S., & Salzberg, S. (1993). A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning, 10*, 57-78.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory, 13*(1), 21-27.

Dain, O., & Cunningham, R. (2001). *Fusing Heterogeneous Alert Streams into Scenarios* (Vol. 6). Springer. doi:10.1007/978-1-4615-0953-0_5

Daniel, B. (01 de 04 de 2006). *OSSEC*. Obtenido de www.ossec.net

Daniel, B., & Sushil, J. (2002). *Applications of Data Mining in Computer Security* (Vol. 6). Springer US. doi:10.1007/978-1-4615-0953-0

Dash, M., & Liu, H. (24 de January de 1997). Feature Selection for Classification. *Intelligent Data Analysis, 1*(1-4), 131-156. doi:10.1016/S1088-467X(97)00008-5

Davison, A., & Hinkley, D. (1997). *Bootstrap methods and their application.* Cambridge: Cambridge University Press.

De la Hoz Franco, E., De la Hoz Correa, E., Ortiz Garcia, A., Ortega Lopera, J., & Martinez Alvarez, A. (2014). Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps. *Knowledge-Based Systems, 71*, 332-338.

De la Hoz, E., Ortiz, A., Ortega, J., & De la Hoz, E. (2013). Network Anomaly Classification by Support Vector Classifiers Ensemble and Non-Linear Projection Technique. *HAIS - Hybrid Artificial Intelligent Systems.* Salamanca, España.

Debar, H., Dacier, M., & Wespi, A. (Julio-Agosto de 2000). A Revised Taxonomy for Intrusion-Detection Systems. *Springer, 55*(7-8), 361-378. doi:10.1007/BF02994844

Dempster, A., Lair, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 39*(1), 1-38.

Denison, D., Mallick, B., & F.M. Smith, A. (1998). A Bayesian CART Algorithm. *Biometrika, 85*(2), 363-377.

Devijver, P. (Abril de 1977). Reconnaissance des Formes par la Méthode des Plus Proches Voisins. *Doctoral Dissertation*. Paris, Italia: Univ. de París VI.

Devijver, P., & Kittler, J. (1982). *Pattern recognition : a statistical approach.* New York, Englewood Cliffs, USA: Prentice/Hall International.

Devyver, P. A., & Kittler, J. (1982). *Pattern Recognition: A Statistical Approach.* Michigan, USA: Prentice-Hall.

Dittenbach, M., Merkel, D., & Rauber, A. (2000). The growing hierarchical self-organizing map. *Proceedings of the international joint conference on neural networks, VI*, 15-19.

Doak, J. (1992). An evaluation of feature-selection methods and their application to computer security. Tech. rep., University of California, Department of Computer Science.

Duda, R., Hart, P., & Stork, D. (1996). Pattern Classification and Scene Analysis: Part I Pattern Classification. En *Pattern Classification and Scene Analysis.* John Wiley & Sons.

Duin, R. (2000). Classifiers in almost empty spaces. *IEEE Explore*.

Duran, F. F., Martinez Sanchez, I., & Sanchez Meraz, M. (2015). Improving Informatics Security Using Quality Control Circles. *PROCEEDINGS OF THE 22015 THIRTY FIFTH CENTRAL AMERICAN AND PANAMA CONVENTION*, 1-5.

Eckmann, S. (2001). *http://citeseerx.ist.psu.edu/.* Obtenido de http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.16.4366&rep=rep1&type=pdf

Elemento, O. (1999). Apport de l'analyse en composantes principales pour l'initialisation et la validation de cartes de kohonen. Inria Nancy - Grand Est: INRIA.

Eskin, E., Arnold, A., Prerau, M., Portnoy, L., & Stolfo, S. (2002). A geometric framework for unsupervised anomaly detection: detecting intru-sions in unlabeled data. *Applications of Data Mining in Computer Security*.

Everett, D. (1992). *Identity Verification and Biometrics.* Boca Raton, FL, USA: CRC Press, Inc.

Fanglu, G., Chen, J., & Chiueh, T. (2006). Spoof Detection for Preventing DoSv Attacks against DNS Servers|. *En 26th IEEE International Conference*, 37-47.

Fano, U. (15 de Diciembre de 1961). Effects of Configuration Interaction on Intensities and Phase Shifts. *Physical Review, 124*(6), 1866-1878.

Fawcett, T. (2006). An introduction to ROC analysis. (J. Elsevier, Ed.) *Pattern Recogn Letters, 27*(8), 861-874.

Feldmeier, D., & Karn, P. (1989). UNIX Password Security - Ten Years Later. *citeseer.ist.psu.edu/188968.html*, 44-63. Obtenido de citeseer.ist.psu.edu/188968.html

Fernandes, S., Kamienski, C., Kelner, J., Mariz, D., & Sadok, D. (9 de October de 2008). A stratified traffic sampling methodology for seeing the big picture. *Computer Networks, 52*(14), 2677-2689. doi:doi:10.1016/j.comnet.2008.05.011

Fix, E., & Hodges, J. (1951). *Discriminatory analysis, nonparametric discrimination consistency properties.* Technical Report 4, US Air Force, School of Aviation Medicine. Randolph Field, TX.

Fix, E., & Hodges, J. (1951). *Discriminatory analysis. Nonparametric estimation: Consistency properties.* University of California, Berkeley. Randolph Field, Texas: University of California.

Fix, E., & Hodges, J. (1952). *Discriminatory analysis, nonparametric discrimination: small sample performance.* Technical Report 11, US Air Force, School of Aviation Medicine, Randolph Field, TX.

Fleuret, F. (5 de December de 2004). Fast binary feature selection with conditional mutual information. (I. Guyon, Ed.) *Journal of Machine Learning Research*, 1531–1555.

Foithonga, S., Pinngernb, O., & At, B. (2012). Feature subset selection wrapper based on mutual information and rough sets. *Expert Systems with Applications, 39*(1), 574–584. doi:doi:10.1016/j.eswa.2011.07.048

Forgy, E. (1965). Cluster analysis of multivariate data: efficiency vs interpretability of classifications. *Biom 21*, 768-769.

Friston, K., Ashburner, J., Kiebel, S., Nichols, T., & Penny, W. (2007). *Statistical Parametric Mapping: The Analysis of Functional Brain Images.* Elsevier.

Fritzke, B. (1995). A growing neural gas network learns topologies. (G. Tesauro, D. Touretzky, & T. Leen, Edits.) *Advances in Neural Information Processing Systems 7*, 625-632.

Fukunaga , K. (1990). *Introduction to Statistical Pattern Recognition* (2 ed.). (W. Rheinboldt, Ed.) New York, USA: Academic Press.

Fyodor. (01 de 04 de 2015). *Network Mapping Tool*. Obtenido de http://www.insecure.org/nmap

Geisser, S. (1993). *Predictive inference: An Introduction.* Minnesota: Chapman & Hall, Inc. doi:10.1007/978-1-4899-4467-2

Ghorbani, A., Lu, W., & Tavallae, M. (2009). *Network Intrusion Detection and Prevention: Concepts and Techniques.*

Ghorbani, A., Lu, W., & Tavallaee, M. (2010). Evaluation Criteria. Network Intrusion Detection and Prevention. Concepts and Techniques. Advances in Information Security. *Springer US*, 161-183.

Ghosh, J. (2002). Multiclassifier systems: Back to the future. *MCS '02: Proceedings of the Third International Workshop on Multiple Classifier Systems*, 1-15.

Girardin, L. (1999). *An Eye on Network Intruder-Administrator Shootouts.* Santa Clara, California, Estados Unidos de America.

Gómez, J., Gil, C., Baños, R., López Márquez, A., Montoya, F., & Gil Montoya, M. (2013). A Pareto-based multi-objective evolutionary algorithm for automatic rule generation in network intrusion detection systems. *Soft Computing*, *17*(2), 255-263.

Gong, F. (2003). Deciphering detection techniques: Part ii. *Anomaly-based intrusion detection McAfee Network Security Technologies Group, White paper, 1*, 1-10.

Graf, H., Cosatto, E., Bottou, L., Durdanovic, I., & Vapnik, V. (2005). Parallel support vector machines: The Cascade svm. *Advances in Neural Information Processing Systems*, 521-528.

Guoliang, T., Kaiwang, N., & Ming, T. (15 de June de 2008). EM-type algorithms for computing restricted MLEs in multivariate normal distributions and multivariate t-distributions. *Computational Statistics and Data Analysis, 52*(10), 4768-4778. doi:DOI: 10.1016/j.csda.2008.03.022

Harrald, J., Schmitt, S., & Shrestha, S. (2004). The Effect of Computer Virus Occurrence and Virus Threat Lever on Antivirus Companies. *Engineering Management Conference, IEEE*, 780-784.

Haykin, S. (1999). *Neural networks* (2 ed.). Prentice-Hall.

He, J., Lan, M., Tan, C., Sung, S., & Low, H. (2004). Initialization of cluster refine-ment algorithms: A review and comparative study. *Proceedings of International Joint Conference on Neural Networks (IJCNN)*.

Heady, R., Luger, G., Maccabe, A., & Servilla, M. (1990). The Architecture of a Network Level Intrusion Detection System. *Technical report, Department of Computer Science, University of New Mexico*.

Hellman, M. (1970). The Nearest Neighbor Classification Rule with a Reject Option. *Systems Science and Cybernetics, IEEE Transactions on Systems, 6*(3), 179 - 185. doi:10.1109/TSSC.1970.300339

Hellman, M., & Raviv, J. (Julio de 1970). Probability of Error, Equivocation, and the Chernoff Bound. *IEEE Transactions On Information Theory, 16*(4), 368-372.

Heskes, T. (2001). Self-Organizing Maps, Vector Quantization, and Mixture Modeling. *IEEE TRANSACTIONS ON NEURAL NETWORKS, 12*(6), 1299 - 1305. doi:10.1109/72.963766

Hilera González, J., & Martínez Hernando, V. (2000). *Redes neuronales artificiales: fundamentos modelos y aplicaciones.* Madrid: Alfaomega Ra-Ma.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedingns of the National Academiy of Sciences, 79*(8), 2554-2558.

Huerta, A. (01 de 04 de 2002). *Seguridad en Unix y redes.* Obtenido de https://www.rediris.es/cert/doc/unixsec/unixsec.pdf

Inteco. (01 de 05 de 2015). *Instituto Nacional de Tecnólogias de la Comunicación*. Obtenido de https://www.incibe.es/

John, G., Kohavi, R., & Pfleger, K. (1994). Irrelevant features and the subset selection problem. En a. a. Journal version in AIJ (Ed.), *International Conference on Machine Learnig* (págs. 121-129). http://csxstatic.ist.psu.edu/about. Obtenido de http://citeseer.ist.psu.edu/john94irrelevant.html

Juan, A., & Vidal, E. (2000). 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, 1876*, 842-852.

Kalyanmoy, D. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms.* NY, USA: Wiley.

Kandeeban, S. S., & Rajesh, R. S. (2010). Integrated Intrusion Detection System Using Soft Computing. *International Journal of Network Security*, 87-92.

Kaur, R., Kumar, G., & Kumar, K. (2015). A Comparative Study of Feature Selection Techniques for Intrusion Detection. *2nd International Conference on Computing for Sustainable Global Development* (págs. 2120-2124). IEEExplore Digital Library.

Kayacık, H., Zincir-Heywood, A., & Heywood, M. (2005). Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets. *Proceedings of the 3rd Conference on Privacy, Security and Trust.*

Kayacik, H., Zincir-Heywood, A., & Heywood, M. (4 de Junio de 2007). A hierarchical SOM-based intrusion detection system. *Engineering Applications of Artificial Intelligence, 20*, 439–451. doi:10.1016/j.engappai.2006.09.005

Kendall, K. (1998). A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems. *Massachusetts Institute of Technology Master´s thesis*.

Kendall, K. (1998). A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems. Massachusetts: Massachusetts Institute of Technology Master's Thesis.

Kira, K., & Rendell, L. (1992). The feature selection problem: traditional methods and a new algorithm. *Proceedings of the 2nd Workshop on Hot Topics in Networks (HotNets-II), AAAI Press*, (págs. 129–134). Los Angeles, California, USA.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. 2 (12)*, págs. 1137-1143. San Francisco: Morgan Kaufmann, Montreal.

Kohavi, R., & John, G. (1997). Wrappers for features subset selection. *Artificial Intelligence - Special issue on relevance*, 273-324.

Kohl, J., Neuman, B., & Ts'o, T. (1994). The Evolution of the Kerberos Authentication Services. *IEEE Computer Society Press*, 79-94.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics, 43*(1), 59-69|.

Kohonen, T. (1990). The Self-Organizing Map. *Proceedings of the IEEE, 78*(9), 1464-1480.

Kohonen, T. (2001). *Self-Organizing Maps* (3 ed., Vol. 30). Springer-Verlag Berlin Heidelberg. doi:10.1007/978-3-642-56927-2

Kohonen, T. (2001). *Self-Organizing Maps.* Springer.

Kotzanikolaou, P., & Douligeris, C. (2007). Computer Network Security: Basic Background and Current Issues. En P. Kotzanikolaou, & C. Douligeris, *Network Security:Current Status and Future Directions* (págs. 1-12). Wiley-IEEE Press.

Kreibich, C., & Crowcroft, J. (2003). Honeycomb-creating intrusion detection signatures using honeypots. *Proceedings of the 2nd Workshop on Hot Topics in Networks (HotNets-II).*

Kumar, S., & Spafford, E. (1995). A Software Architecture to Support Misuse Intrusion Detection. *Proceedings of the 18th National Information Security Conference.*

Lakshmanan, V., Fritz, A., Smith, T., Hondl, K., & Stumpf, G. (2007). An automated technique to quality control radar reflectivity data. *Journal of applied meteorology and climatology, 46*(3), 288-305.

Lazarevic, A., Kumar, V., & Srivast, J. (2005). *Intrusion Detection: A Survey* (Vol. 5). US: Springer US. doi:10.1007/0-387-24230-9_2

Lazarevic, A., Kumar, V., & Srivastava, J. (2005). Intrusion Detection: A survey. En V. Kumar, J. Srivastava, & A. Lazarevic, *Managing Cyber Threats* (págs. 19-78). Minnesota, United States of America: Springer.

Levin, I. (2000). KDD-99 classifier learning contest, LLSoft´s results overview. *SIGKDD Explorations, 1*(2), 67-75.

Lidong, Z., & Haas, Z. (2002). Securing ad hoc networks. (IEEE, Ed.) *Network, IEEE, 13*(6), 24-30. doi:10.1109/65.806983

Liu, Y. (14-16 de September de 2004). A hybrid neural network learning system. *Computer and Information Technology, 2004. CIT '04*, 1016 - 1021. doi:10.1109/CIT.2004.1357329

LL-MIT. (2014). *Publications*. Recuperado el 26 de June de 2015, de Lincoln Laboratory of Massachusetts Institute TecnologyLincoln Laboratory of Massachusetts Institute Tecnology: http://www.ll.mit.edu/publications/index.html

López, M., Ramírez, J., Górriz, J., Álvarez, I., Salas González, D., Segovia, F., . . . Gómez Río, M. (8 de March de 2011). Principal component analysis-based techniques and supervised

classification schemes for the early detection of Alzheimer's disease. *Neurocomputing, 74*(8), 1260-1271.

Lotlikar, R., & Kothari, R. (1999). Multilayer perceptron based dimensionality reduction. *Neural Networks, IJCNN '99. International Joint Conference, 3*, 1691 - 1695. doi:10.1109/IJCNN.1999.832629

Lunt, T. (1990). IDES: an intelligent system for detecting intruders. *Computer Security, Threat and Countermeasures.* Rome.

Lunt, T., & Jagannathan, R. (1988). A Prototype Real-Time Intrusion-Detection Expert System. *Security and Privacy, IEEE Symposium*, (págs. 1-59).

Luttrell, S. (1989). Hierarchical self-organising networks. *Artificial Neural Networks, 1989., First IEE International Conference on (Conf. Publ. No. 313)*, 2-6.

M. Borghi, M., Maggiolino, M., L. Montagnani, M., & Nuccio, M. (2012). Determinants in the online distribution of digital content: an exploratory analysis. *European Journal for Law and Technology, 3*(2).

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. (L. N. Cam, Ed.) *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1*, 281–297.

Maxion, R., & Roberts, R. (2004). *Proper Use of ROC Curves in Intrusion/Anomaly Detection.* Technical Report CS-TR-871, Uni-versity of Newcastle upon Tyne, School of Computing Science.

Mbareen, S., Vaughn, R., & Bridges, S. (2004). Intrusion Sensor Data Fusion in an Intelligent Intrusion Detection System Architecture. *Proceedings of the 37th Annual Hawaii International Conference*, (pág. 10).

Mu-Chun SU, T., & Chang, H. (2002). Improving the self-organizing feature map algorithm using an efficient initialization scheme. *Tamkang Journal of Science and Engineering, 5*, 35-48.

Muller, K.-R., Smola, A., Ratsch, G., Scholkopf, J., & Vapnik, V. (s.f.). Using support vector machines for time series prediction.

Naiqi, W., Qian, Y., & Chen, G. (2006). A Novel Approach to Trojan Horse Detection by Process Tracing. *Proceedings of the 2006 IEEE International Conference*, 721-726.

Navidi, W. (2014). *Statistics for Engineers and Scientists 4th Edition.* McGraw-Hill Education.

NIST. (01 de 04 de 2015). *National Institute of Standards and Technology*. Obtenido de http://www.nist.gov/

Noel, S., Wijesekera, D., & Youman, C. (2002). Modern intrusion detection, data mining, and degrees of attack guilt. *Center for Secure Information Systems*. George Mason University. Obtenido de Securing the World's Cyber Infrastructure: http://csis.gmu.edu/noel/pubs/IDS_chapter.pdf

Northcutt, S., Winters, S., Kent, K., & Ritchey, R. (2005). *Inside Network Perimeter Security: An Analyst Handbook* (Second Edition ed.).

NSL-KDD. (s.f.). Obtenido de http://www.iscx.ca/NSL-KDD/

Ocsa, A., Bedregal, C., & Cuadros-Vargas, E. (12-17 Aug. de 2007). DB-GNG: A constructive self-organizing map based on density. *Proceedings of the International Joint Conference on Neural Networks (IJCNN07)*, 1953-1958. doi:10.1109/IJCNN.2007.4371257

Odgaard, P., & Wickerhauser, M. (9-13 de July de 2007). Karhunen-Loeve (PCA) based detection of multiple oscillations in multiple measurement signals from large-scale process plants. *American Control Conference*, 5893 - 5898. doi:10.1109/ACC.2007.4282149

Olovsson, T. (1992). *A Structured Approach to Computer Security.* Chalmers University of Technology.

Ortiz, A., Ortega, J., Díaz, A., & Prieto, A. (2011). Network Intrusion Prevention by Using Hierarchical Self-Organizing Maps and Probability-Based Labeling. En S. B. Heidelberg (Ed.), *Advances in Computational Intelligence. 11th International Work-Conference on Artificial Neural Networks, IWANN* (págs. 232-239). Torremolinos-Málaga, Spain: Lecture Notes in Computer Science.

Pai, P.-F., & Hong, W.-C. (2005). Support vector machines with simulated annealing algorithms in electricity load forecasting. *Energy Conversion and Management, 46*(17), 2669-2688.

Panda, M., Abraham, A., & Patra, M. (2010). Discriminative multinomial naïve Bayes for network intrusion detection. En IEEE (Ed.), *6th Conference on Information Assurance and Security (IAS)*, (págs. 5-10).

Pena, J., Lozano, J., & Larranaga, P. (1999). An empirical comparison of four ini-tialization methods for the k-means algorithm. *Pattern Recogn, 20*, 1027-1040.

Pfahringer, B. (2000). Winning the FDD99 classification cup: bagged-boosting. *SIGKDD Explorations, 1*(2), 65-66.

Powell, D., & Stroud, R. (2001). *Conceptual Model and Architecture, Deliverable D2, Project MAFTIA IST-1999-11583.* Zurich: IBM Zurich Research Laboratory Research Report RZ 3377.

RAE. (01 de 04 de 2015). *Real Academia Española*. Obtenido de http://lema.rae.es/drae/?val=seguridad

RAE. (01 de 04 de 2015). *Real Academia Española*. Obtenido de http://lema.rae.es/drae/?val=seguro

RAE. (01 de 04 de 2015). *Real Academia Española*. Obtenido de http://lema.rae.es/drae/?val=informat%C3%ADca

RAE. (01 de 04 de 2015). *Real Academia Española*. Obtenido de http://lema.rae.es/drae/?val=anomal%C3%ADa

Raudys, S., & Jain, A. (Marzo de 1992). Small sample size efects in statistical pattern recognition: recommendations for practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*(3), 252-264.

Reeves, C., & Singh Billan, G. (2001). Using Decision Surface Mapping in the Automatic Recognition of Images. En *Artificial Neural Nets and Genetic Algorithms* (págs. 82-85). Springer Vienna. doi:10.1007/978-3-7091-6230-9_19

Refaeilzadeh, P., Tang, L., & Lui, H. (6 de Noviembre de 2008). k-fold Cross-Validation. Arizona State University.

Richards, J., & Jia, X. (2006). *Remote Sensing Digital Image Processing: An Introduction* (4th Edition ed.). Berlin Heidelberg, Germany: Springer-Verlag. Obtenido de springeronline.com

Riveiro, M., Johansson, F., Falkman, G., & Ziemke, T. (2008). Supporting maritime situation awareness using self organizing maps and Gaussian mixture models. *Proceedings of the 2008 Conference on 10th Scandinavian Conference on Artificial Intelligence (SCAI 2008), 1*, págs. 84-91.

Roesch, M. (7-12 de November de 1999). Snort-Lightweight Intrusion Detection for Networks. *Proceedings of LISA '99: 13th Systems Administration Conference*, 229-238.

Roesch, M. (2005). *Lightweight Intrusion Detection for Networks.* Obtenido de www.snort.org

Rubio, G., Guillen, A., Herrera, L., Pomares, H., & Rojas, I. (2008). Use of specific-to-problem kernel functions for time series modeling. *ESTSP'08: Proceedings of the European Symposium on Time Series Prediction*, 177-186.

Rusell, D., & Gangemi, G. (1991). *Computer Security Basics.* California: O‟Reilly & Associates, Inc., Sebastopol.

Saâdaoui, F. (2010). Acceleration of the EM algorithm via extrapolation methods: Review, comparison and new methods. *Computational Statistics & Data Analysis, 54*(3), 750-766.

Sadkhan, S. (2009). On artificial intelligence approaches for network intrusion detection systems. *MASAUM Journal of Computing*, 236-243.

Samad, T., & Harp, S. (1992). Self-Organization with Partial Data. *Network*, 205-212.

Sandeep, K. (1995). *Classification and Detection of Computer Intrusions.* citeseer.ist.psu.edu/kumar95classification.html. Obtenido de Purdue University.

SANS. (2015). *SANS*. Obtenido de http://www.sans.org/security-resources/idfaq/

Sapkal, S., Kakarwal, S., & Revankar, P. (13-15 de December de 2007). Analysis of Classification by Supervised and Unsupervised Learning. *Conference on Computational Intelligence and Multimedia Applications, 1*, 280 - 284. doi:10.1109/ICCIMA.2007.237

Scholkopf, B., & Smola, A. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* Cambridge, MA, USA: MIT Press.

Schott, J. (Diciembre de 1998). Estimating correlation matrices that have common eigenvectors. *Computational Statistics & Data Analysis*(27), 445-459.

Schwartz, S., & Carpenter, K. (August de 1999). The right answer for the wrong question: consequences of type III error for public health research. *Am J Public Health, 89*(8), 1175–1180. doi:10.1007/978-1-4899-4467-2

Schweitzer, F. (1997). *Self-Organization of Complex Structures: from individual to collective dynamics.* Berlin: CRC Press.

Security, I. f. (01 de 04 de 2015). *Institute for Internet Security*. Obtenido de http://www.internet-sicherheit.de/en/research/recent-projects/internet-early-warning-systems/internet-analysis-system/recent-results/

Smith, L. (2002). Tutorial on Principal Components Analysis.

Spafford, E. (1989). Crisis and Aftermath. *Communications of the ACM*, 678-687.

SRI. (s.f.). *SRI International*. Obtenido de http://www.sri.com/

Strehl, A., & Ghosh, J. (2002). Cluster ensembles – a knowledge reuse framework for combining partitionings. *Proceedings of AAAI2002*, 93-98.

Tasdemir, K., Milenov, P., & Tapsall, B. (March de 2011). Topology-based hierarchical clustering of self-organizing maps. *IEEE Trans Neural Netw, 22*(3), 474-485. doi:10.1109/TNN.2011.2107527.

Tatsuoka, M. (Junio de 1974). Multivariante Analysis: Techniques for Educational and Psychological Research. *39*(2), 269-274.

Tavallaee, M., Stakhanova, N., & Ghorbani, A. (2010). Toward credible evaluation of anomaly-based intrusion-detection methods. *IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews*, 516-524. doi:10.1109/TSMCC.2010.2048428

Tay. (2001). Application of support vector machines in financial time series forecasting. *Omega: The International Journal of Management Science, 29*(4), 309-317.

Theodoridis, S., & Koutroumbas, K. (2009). *Pattern Recognition.* Burlington , USA: Academic Press - Elsevier .

Theodoridis, S., & Koutroumbas, K. (2009). *Pattern Recognition, 4th Edition.* Elsevier Inc.

Tipton, H., & Krause, M. (2006). *Information Security Management Handbook* (Vol. 5). Auerbach Publications.

Turk, M., & Pentland, A. (3-6 Jun 1991). Face recognition using eigenfaces. *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*, (págs. 586 - 591). doi:10.1109/CVPR.1991.139758

University of California. (28 de October de 1999). (Information and Computer Science, University of California. Irvine, CA 92697-3425.) Obtenido de http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

University of California. (28 de October de 1999). *KDD Cup 1999 Data*. (Irvine) Recuperado el 15 de Agost de 2015, de The UCI KDD Archive: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Vapnik, V. (1998). *Statistical Learning Theory.* New York: John Wiley and Sons, Inc.

Vesanto, J., Himberg, J., Alhoniemi, E., & Parhankangas, J. (2000). *SOM toolbox.* Helsinki University of Technology. Finland: Helsinki University of Technology.

Vesanto, J., Himberg, J., Alhoniemi, E., & Parhankangas, J. (April de 2000). *SOM Toolbox for Matlab 5.* Report A57, Laboratory of Computer and Information Science (CIS). Recuperado el 2016, de http://www.cis.hut.fi/projects/somtoolbox/

VIM, W. G.—B. (2008). *Bureau International des Poids et Mesures.* Recuperado el 26 de Junio de 2015, de Common Documents: http://www.bipm.org/utils/common/documents/jcgm/JCGM_200_2008.pdf

Wang Ko, C. C. (1996). Execution Monitoring of Security Critical Programs in a Distributed System: A Specification-Based Approach. *Dissertation Doctor of Philosophy*.

Wang, H., & Hu, Z. (22 de October de 2009). On EM Estimation for Mixture of Multivariate t-Distributions. *Neural Processing Letters*, 243-256. doi:10.1007/s11063-009-9121-5

Wenli, L., Xiaolong , Z., Tao, W., & Hiu, W. (2014). Collaboration Pattern and Topic Analysis on Intelligence and Security Informatics Research. *INTELLIGENCE AND SECURITY INFORMATICS*, 39-45.

Wu, S. X., & Banzhaf, W. (2010). The use of computational intelligence in intrusion detection systems: A review. *Applied Soft Computing*, 1-35.

Wu, S., & Banzhaf, W. (2010). The use of computational intelligence in intrusion detection systems: A review. *Applied Soft Computing*, 1-35.

Wu, W., Massart, D., & Jong, S. (1997). The kernel pca algorithms for wide data part i: Theory and algorithms. *Chemometrics and Intelligent Laboratory Systems, 36*(2), 165-172.

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

Zargari, S., & Voorhis, D. (2012). Feature Selection in the Corrected KDD-dataset. *EIDWT '12 Proceedings of the 3rd International Conference on Emerging Intelligent Data and Web Technologies*, (págs. 174-180).

Zhang, D.-Q., & Chen, S.-C. (2003). Clustering incomplete data using kernel-based fuzzy c-means algorithm. *Neural Process, 18*(3), 155-162.

Ziolko, S., Weissfeld, L., Klunk, W., Mathis, C., Hoge, J., Lopresti, B., . . . Price, J. (2006). Evaluation of voxel-based methods for the statistical analysis of PIB PET amyloid imaging studies in Alzheimer's disease. *NeuroImage, 33*(1), 94-102.

Zseby, T. (2003). Stratification Strategies for Sampling-based Non-intrusive Measurement of One-way Delay. *Proceedings of Passive and Active Measurement Workshop*, 171-179.

Lippmann, R., Haines, J., Fried, D., Korba, J. and Das, K. (2000). The 1999 DARPA off-line intrusion detection evaluation. *Computer Networks*, 34(4), pp.579-595.