Institución Universitaria

Clasificador híbrido multinivel basado en Rough Fuzzy C-Means para fortalecer la robustez de un sistema de detección de intrusos

Juan David Grajales Bustamante

INSTITUTO TECNOLÓGICO METROPOLITANO
Facultad de Ingenierías
Medellín, Colombia
2019

Clasificador híbrido multinivel basado en Rough Fuzzy C-Means para fortalecer la robustez de un sistema de detección de intrusos

Juan David Grajales Bustamante

Tesis presentada como requisito para optar al título de:
Magíster en Seguridad Informática

Director:
Prof. Edilson Delgado Trejos, Ing, MSc, PhD

Grupo de Investigación: Calidad, Metrología y Producción (CM&P)
Línea de Investigación: Calidad y Metrología
Laboratorio: Análisis de Medición y Soporte de Decisión (AMYSOD)
Área: Softmetrología para Seguridad Informática

INSTITUTO TECNOLÓGICO METROPOLITANO
Facultad de Ingenierías
Medellín, Colombia
2019

Multilevel Hybrid Classifier Based on Rough Fuzzy C-Means for Enhancing the Robustness of an Intrusion Detection System

Juan David Grajales Bustamante

Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of:
Master's in Information Security

Supervisor:
Prof. Edilson Delgado-Trejos, Ing, MSc, PhD

Research Group: Quality, Metrology and Production (CM&P)
Research Line: Quality and Metrology
Research Lab: Measurement Analysis and Decision Support (AMYSOD)
Subject: Softmetrology for Information Security

INSTITUTO TECNOLÓGICO METROPOLITANO
Facultad de Ingenierías
Medellín, Colombia
2019

Dedicatoria

Agradezco a Dios por bendecirme, por guiarme a lo largo de mi existencia, a mi incondicional y amada esposa por su incesante apoyo y dedicación en esas arduas noches de desvelo para no desfallecer. Con su gran amor a lo largo de toda esta travesía poderla culminar con gran satisfacción por todo el esfuerzo realizado.

a mi familia por su inmensurable comprensión y amor.

Agradecimientos

Agradezco a mi director Edilson Delgado Trejos por sus oportunas, constantes y siempre acertadas orientaciones, por su permanente paciencia y confianza para realizar este trabajo.

A Juan Pablo mi amigo, por su oportuna colaboración y apoyo incondicional para poder terminar este trabajo.

A la profesora Diana Orrego por sus apropiadas e inteligentes recomendaciones, por su don de servir y ayudar a las demás personas.

Resumen

Las redes computacionales se han convertido en una parte esencial para las organizaciones, donde el tráfico electrónico se convierte en un soporte vital de las empresas y la inversión en esta área incrementa el potencial de la capacidad de negociación. En este contexto, la ciberseguridad es un desafío dado el panorama de amenazas en constante evolución y toma gran relevancia no sólo en la comunidad científica, sino a nivel empresarial y gubernamental. Una gran variedad de Sistemas de Detección de Intrusos (IDS) basados en Aprendizaje de Máquinas, han jugado un papel importante en la detección de anomalías estructurados en la clasificación. Sin embargo, estas técnicas siguen siendo débiles en el manejo y modelado de los ataques y amenazas, dada la complejidad de las redes computacionales y el comportamiento no lineal de los ataques, lo cual provoca que sean impredecibles a lo largo del tiempo, generando incertidumbres que constituyen un ensamble interactuante y multivariado. En este contexto se estudiarán los diferentes métodos de detección de intrusos como son: Detección basada en Conocimiento, Detección basada en Métodos Estadísticos y Detección basada en Aprendizaje de Máquinas, donde se discuten. las técnicas de análisis supervisado, no supervisado y semi supervisado. Por consiguiente se presenta un esquema de clasificación híbrido multinivel, que incluye rutinas *Rough Fuzzy c-Means*, con el objetivo de lograr robustez en un IDS para cuatro tipos de ataques: *Denial of Service (DoS), Probing Attack (PA), Remote to Local (R2L)* y *User to Root (U2R)*. Se realizó un proceso de selección de características con los algoritmos *Fuzzy Rough Sets (FRS)* y *Relief F*, encontrando que *Relief F* mejora significativamente la capacidad discriminante del clasificador multinivel. El IDS fue implementado usando la base de datos KDD Cup 99 a través de tres niveles y seis clasificadores. En el nivel uno, el primer clasificador identifica dos clases y una agrupación: DoS, PA y la agrupación compuesta por las clases R2L, U2R y Normal denominada (RUN). El segundo nivel está compuesto por tres clasificadores: los dos primeros identifican las formas de ataque correspondientes a DoS y PA, respectivamente, mientras el tercer clasificador identifica las clases R2L, U2R y Normal. Asimismo, el nivel tres queda compuesto por dos clasificadores que identifican las formas de ataque para R2L y U2R, respectivamente. Se evaluaron tres tipos de clasificadores: Máquinas de Vectores de Soporte (SVM), k- Vecinos más Cercanos (k-NN) y *Fuzzy c-Means* Semi-Supervisado (SSFCM). Se encuentra que SVM ofrece el mejor desempeño en el nivel uno y los dos primeros clasificadores del nivel dos, k-NN en el último clasificador del nivel dos y SSFCM en los clasificadores del nivel tres. Finalmente, se lograron errores inferiores al 1 % en los niveles uno y dos, mientras en el nivel tres (clasificación entre las diferentes formas de ataque para R2L y U2R), se obtuvieron errores de alrededor del 7 %. De esta manera, con una precisión global del clasificador en los niveles uno y dos de un 99.69 %, y para el nivel 3, un resultado del 93 % de ataques clasificados como ataques, donde se tomó como función objetivo la robustez del sistema de detección de intrusos.

Palabras clave: Aprendizaje de Máquina, Ataques Informáticos, Clasificación Multi-

nivel, Redes Computacionales, Selección de Características, Sistema de Detección de Intrusos.

Abstract

Computer networks have become an essential part of organizations, where electronic traffic is a fundamental support for companies, and investment in this area increases the negotiation capacity. In this context, cybersecurity is a challenge not only for the scientific community but also businesses and government, as threats are in constant evolution. A variety of Intrusion Detection Systems (IDS) based on Machine Learning have played an important role in the detection of structured anomalies in the classification. However, these techniques are still weak in handling and modeling attacks and threats given the complexity of computer networks and the non-linear behavior of attacks, which causes them to be unpredictable over time, generating uncertainties that constitute an interacting and multivariate assembly. In this context, the different methods of intrusion detection will be studied, such as: Knowledge-based detection, Detection based on Statistics and Detection based on Machine Learning, where they are discussed. Supervised, unsupervised and semi-supervised analysis techniques. Therefore, a multi-level hybrid classification scheme that includes Rough Fuzzy c-Means routines is presented, with the aim of achieving robustness in an IDS for four types of attacks: *Denial of Service (DoS), Probing Attack (PA) , Remote to Local (R2L)* and textit User to Root (U2R). A process of feature selection was performed with the Fuzzy Rough Sets (FRS) and Relief F algorithms, finding that Relief F significantly improves the discriminant capacity of the multilevel classifier. The IDS was implemented using the KDD Cup 99 database through three levels and six classifiers. In level one, the first classifier identifies two classes and one grouping: DoS, PA and the grouping composed of classes R2L, U2R and Normal. The second level is composed of three classifiers, the first two identify the attack forms corresponding to DoS and PA, respectively, while the third classifier identifies the classes R2L, U2R and Normal. Likewise, level three is composed of two classifiers that identify the attack forms for R2l and U2R, respectively. Three types of classifiers were evaluated: Vectortial Support Machines (SVM), k- Nearest Neighbors (k-NN) and Fuzzy c-Means Semi-Supervised (SSFCM); finding that SVM offers the best performance in level one and the first 2 classifiers of level two, k-NN in the last classifier of level two and SSFCM in classifiers of level three. Finally, errors lower than 1 % were obtained in levels 1 and 2, while in level 3 (classification between the different attack forms for R2L and U2R), errors of around 7 % could be obtained. In this way, the hybrid multilevel classification scheme was validated, with an overall accuracy of the classifier at level one and two of 99.69 % for level 3 with 93 % of attacks classified as attacks, taking as objective the robustness of the intrusion detection system.

keywords: Machine Learning, Computer Attacks, Multilevel classification, Computer networks, Selection of Features, Intrusion Detection System.

Contenido

Lista de Figuras

Lista de Tablas

1. Introducción

Las redes computacionales se han convertido en una parte esencial para las organizaciones, ya que el tráfico electrónico se convierte en un soporte vital de las empresas, por lo que la inversión en esta área incrementa el potencial de la capacidad de negociación. Esto quiere decir, que cada vez hay mayor exigencia en la seguridad informática, a fin de proteger los datos que se encuentran en la red, específicamente, se requiere de sistemas de Detección de Intrusos (IDS, por sus siglas inglés) más exigentes que ayuden a alertar y detectar sobre alguna anomalía en la infraestructura de la red corporativa, como lo es el método de Detección basada en Aprendizaje de Máquina, donde se discuten las técnicas de análisis supervisado, no supervisado y semi supervisado.

1.1. Justificación

El ciberespacio se ha convertido en una fuente indispensable para el intercambio de información entre los usuarios y las organizaciones, dado que elimina barreras geográficas y mejora el acceso de la población a los diferentes sectores y servicios, como la salud, la educación, la industria, el comercio electrónico y las finanzas, entre otras. Considerando que el acceso a la Internet es cada vez más asequible, las personas y las empresas están incrementando su conexión a éste servicio a través de múltiples sistemas computacionales, generando altos volúmenes de tráfico en las redes de datos, lo cual se convierte en una gran preocupación para los profesionales de las Tecnologías de la información (TI).

En este contexto, la seguridad en las redes computacionales ha sido un tema de alto impacto tanto en la comunidad científica como en el sector empresarial. Debido a este auge tecnológico se han venido generando investigaciones y herramientas para proteger usuarios y organizaciones de los diferentes ataques, que se traducen en intrusiones no autorizadas a la red, vulnerando y poniendo en riesgo la privacidad e integridad de la información [1, 2]. Los ataques cibernéticos evolucionan cada día y pueden ser conducidos a través de múltiples vectores y etapas para obtener un mayor porcentaje de efectividad en su objetivo. La gran variedad de intentos de intrusión por parte de individuos con intención maliciosa, logran ataques de tipo “Día cero”, “Denegación de servicio”, “control remoto a un usuario local” y “secuestro de información”, entre otros [3]. Según Symantec [4], el secuestro de los datos, (Ransomware), fue una de las principales amenazas para el año 2018. En la misma lista están los ataques de Denegación de servicio en Internet de las Cosas (IoT por sus siglas en

inglés), al igual que los ataques a los sitios web. En los últimos años tanto los usuarios como las redes han sido víctimas de muchos tipos de ataques. Estos ataques cibernéticos son a veces muy perjudiciales y cuestan miles de millones de dólares cada año [5]. Por lo tanto, los IDS se han desarrollado para detectar los ataques a la infraestructura de las redes computacionales y notificar a los administradores de red. Estos sistemas se están estudiando y estructurando actualmente de una manera más amplia a los nuevos tipos de amenazas, para proporcionar una mayor robustez en el marco de seguridad de la defensa en profundidad de la red computacional [5]. Una gran variedad de Sistemas de Detección de Intrusos basados en Aprendizaje de Máquina, han jugado un papel importante en la detección de anomalías estructurados en la clasificación. Métodos como Redes Neuronales Artificiales, Máquinas de Vectores de Soporte, k-Vecinos más Cercanos, *Fuzzy c-Means*, entre otros [6], detectan amenazas revelando comportamientos sospechosos en un sistema durante un período de tiempo, con alta capacidad para diferenciar entre comportamientos normales y anomalías dentro de una red. Estos métodos, generalmente son utilizados de forma individual o en una sola etapa de clasificación, aunque su precisión y rendimiento no son uniformemente buenos para cada distribución de clase. Razón por la cual, la combinación adecuada de los clasificadores múltiples para cada tipo de distribución de clase en un conjunto de datos, probablemente proporcionaría un resultado altamente preciso. Bajo este contexto, se propone un esquema de clasificación híbrido multinivel que incluya rutinas *Rough Fuzzy c-Means* para lograr robustez en un IDS de redes computacionales.

1.2. Problema

La detección de intrusos es uno de los problemas más importantes en la cyberseguridad, convirtiéndose en un desafío crucial tanto para la comunidad científica como para la industria actual [7]. Con el gran aumento del tráfico en la red, los piratas informáticos y los usuarios malintencionados cada vez más se están ideando nuevas formas de intrusión en la red, por tanto es la tarea más difícil y desafiante en la detección de intrusos distinguir eficazmente entre el tráfico normal y el malicioso [8].

Aunque existen varios enfoques en los métodos de detección de intrusos basados en anomalías para la clasificación, frecuentemente sufren altas tasas de falsos positivos, debido a sus limitaciones para diferenciar el comportamiento de ataque y comportamiento de la conexión normal en evolución [9]. En este sentido, la literatura reporta el uso de diferentes clasificadores individuales o en una sola etapa, como Sistemas Expertos, Modelos Ocultos de Markov, Redes Neuronales Artificiales, Máquinas de Vectores de Soporte, enfoque basado en reglas, k-Vecinos más Cercanos, *Fuzzy c-Means*, entre otros [6], los cuales modelan el comportamiento normal del sistema, bajo la hipótesis de que el comportamiento del atacante difiere del comportamiento normal del usuario, además de ayudar a extraer información útil de grandes

conjuntos de datos [10], sin embargo, la precisión y el rendimiento de los clasificadores de una sola etapa no son uniformemente buenos para cada distribución de clase. Además, no es posible que un solo enfoque de detección proporcione una mejor precisión y reduzca el costo del modelo [11].

El Aprendizaje de Máquina ha jugado un papel muy importante en los Sistemas de Detección de Intrusos (IDS), con diferentes métodos supervisados, no supervisados y semi supervisados, como se describen en el Capítulo 2. No obstante, los métodos de aprendizaje no supervisados a menudo son más desafiantes, ya que los resultados son subjetivos y no hay un objetivo simple para el análisis, como predecir la clase o la variable continua. Estos métodos se realizan como parte del análisis exploratorio de datos [6]. Además de eso, puede ser difícil evaluar los resultados obtenidos de los métodos de aprendizaje no supervisados, ya que no existe un mecanismo universalmente aceptado para realizar su validación [12].

Por otra parte, el alto costo computacional ha sido un problema común para los métodos de Aprendizaje de Máquina. En Árboles de Decisión, las probabilidades de cálculo de diferentes ramas posibles, la determinación de la mejor división de cada nodo y la selección de ponderaciones óptimas para podar los algoritmos contenidos en el árbol, son tareas complicadas e implican un alto costo computacional [13]. Las limitaciones en k-Vecinos más Cercanos para determinar el valor óptimo de k y la métrica de distancia adecuada aumentan el alto costo computacional, especialmente en términos de cálculos de distancia [14]. Así mismo, *Fuzzy c-Means* calcula los grados de membresía en cada iteración, que implica a una operación costosa y otorga un grado de membresía a un punto proporcional de la proximidad a los representantes del grupo. Como consecuencia, el tamaño de la matriz crece como un producto de la cantidad de puntos y agrupaciones, lo que hace que el algoritmo sea computacionalmente costoso para valores altos [15].

Algunos trabajos han adoptado enfoques de construcción híbrida, que permiten al sistema obtener mejores resultados de clasificación, al combinar diferentes técnicas y superar el inconveniente de cada uno, dando como resultado una mayor precisión en la detección de anomalías [16]. Sin embargo, el costo computacional y la complejidad de los sistemas son altos, siguen presentando una mayor tasa de falsos positivos y dejan pasar por alto muchas conexiones de ataque [17], por tanto, la combinación de un grupo de clasificadores en una sola etapa no siempre funciona mejor. Varias combinaciones diferentes posibles de multiclasificadores deben ser validadas [7].

La revisión en la literatura presentada en esta investigación, se enfocó en los métodos de detección de intrusos basados en anomalías para la clasificación, utilizando la base de datos KDD Cup 99, la cual ha sido el conjunto de datos más utilizados en investigaciones para la detección de intrusos con aprendizaje automático y todavía se usa ampliamente. Sin

embargo, en los (165) artículos referenciados no se utilizó la base de datos completa y sin modificaciones, dado que los artículos reportan el uso del 10 % de la base de datos para el entrenamiento del método y la base de datos corregida para la etapa de validación. Se presume que la razón por la que en la literatura no es frecuente encontrar trabajos con la base de datos completa es por su tamaño, dado que cuenta con 4.898.430 muestras de 41 dimensiones.

De acuerdo a lo anterior, se construye la siguiente pregunta de investigación: ¿Un clasificador híbrido multinivel que incluya rutinas *Rough Fuzzy c-Means* puede brindar la robustez suficiente para fortalecer la eficiencia del desempeño de los sistemas de detección de intrusos, superando los inconvenientes derivados de los ensambles interactuantes y multivariados que se forman alrededor de la incertidumbre?

1.3. Hipótesis

Dadas las bondades que ofrecen las estrategias que involucran la conceptualización de conjuntos rugosos *(Rough Sets)* y las ventajas asociadas con las asignaciones a funciones de membresía difusa, se podrá lograr híbridos efectivos en cuanto a robustez a la hora de estructurar sistemas de soporte de decisión. Adicionalmente, mediante la inclusión del poder que tienen los sistemas auto-sintonizables de conformación de clústeres de manera no supervisada, como el *Fuzzy c-Means*, permitirán la conformación de una técnica de identificación de intrusos que supere los inconvenientes dados por los ensambles interactuantes y multivariados que se forman alrededor de la incertidumbre.

1.4. Objetivos

1.4.1. Objetivo general

Proponer un esquema de clasificación híbrido multinivel que incluya configuraciones *Rough Fuzzy C-Means* para lograr robustez en un Sistema de Detección de Intrusos para cuatro tipos de ataques *(Denial of Service -DoS, Probing Attack, Remote to Local -R2L, User to Root -U2R)* en redes computacionales.

1.4.2. Objetivos específicos

- Caracterizar cuatro tipos de ataques *(Denial of Service -DoS, Probing Attack, Remote To Local -R2L, User to root -U2R)*, comunes en redes computacionales, para generar un espacio de representación multivariada.

- Analizar diferentes configuraciones de clasificación híbrida multinivel que incluya rutinas *rough fuzzy c-means*, a fin de detectar y mitigar el mayor número posible de

amenazas que se encuentran en una red de datos.

- Validar el esquema de clasificación híbrido multinivel propuesto, tomando como función objetivo la robustez del sistema de detección de intrusos, a fin de ajustar el desempeño de los algoritmos.

1.5. Organización del manuscrito

Esta tesis inicia en el Capítulo 2 con la revisión bibliográfica de los diferentes métodos de detección de intrusos como son: Detección basada en Conocimiento, Detección basada en Estadísticas y Detección basada en Aprendizaje de Máquinas, donde se discuten las técnicas de análisis supervisado, no supervisado y semi supervisado. En el Capítulo 3, se presenta el marco teórico donde se aborda la conceptualización correspondiente a las redes computacionales, sistemas de detección de intrusos y diferentes tipos de ataque, las técnicas usadas para la construcción de nuevos sistemas de detección de intrusos, como son, esquemas de selección de características (*fuzzy rough sets* y *relief F*), máquinas de vectores de soporte, k-vecinos más cercanos y *fuzzy c-means* para los ejercicios de clasificación y validación. En el Capítulo 4, se presenta el marco experimental, donde se describe la base de datos y se expone el modelo propuesto para la generación de los espacios de representación, la selección de características para reducir la dimensionalidad de los diferentes tipos de ataques, además de la estructuración metodológica que permita la detección de los cuatro tipos de ataques. En el Capítulo 5, se presentan los resultados y discusión de los experimentos realizados. El Capítulo 6 discute las conclusiones y el trabajo futuro que queda de los hallazgos de la tesis. Por último, se listan las referencias bibliográficas y los anexos que complementan la información discutida en el desarrollo del documento.

2. Estado del arte: Métodos de clasificación para la detección de intrusos

Muchos problemas pueden abordarse como una clasificación en *Big Data*, Ciencia de Datos y Aprendizaje Automático [18]. Por otra parte, los Sistemas de Detección de Intrusos descubren anomalías revelando características sospechosas en un sistema durante un período de tiempo. La eficacia de estos sistemas es su capacidad para diferenciar entre comportamientos normales y anomalías dentro de una red [19]. Los IDS basados en anomalías para la clasificación se consideran esenciales para la seguridad de la red debido al incremento en el uso de las Tecnologías de la Información y Comunicación -TICs. Así mismo, los IDS multinivel o de varias etapas mejoran la capacidad de ataques conocidos, desconocidos y falsas alarmas que pueden ser vulnerables cuando se utilizan IDSs con un solo clasificador [20].

El estado del arte se describe en un mapa conceptual presentado en la Figura **2-1**, se fundamenta en los Métodos de clasificación para la Detección de Intrusos basados en anomalías. Las investigaciones actuales se centran principalmente en tres líneas de investigación alrededor de estas técnicas: Sistemas basados en Conocimiento, Sistemas basados en métodos Estadísticos y Sistemas basados en Aprendizaje de Máquinas. Las técnicas de conocimiento que más se reportan en la literatura son: Marcos, Reglas y Sistemas Expertos. En la Detección basada en Estadísticas se encuentran las Paramétricas Univariadas, No Paramétricas Multivariadas, Redes Bayesianas y Modelos Ocultos de Markov. Por último en la Detección basada en Máquinas de Aprendizaje se presentan diferentes métodos: supervisados, no supervisados y Semi Supervisados, para la regresión, clasificación y agrupamiento en la detección de anomalías, donde la literatura reporta explícitamente que todavía hay muchos vacíos para la detección de amenazas.

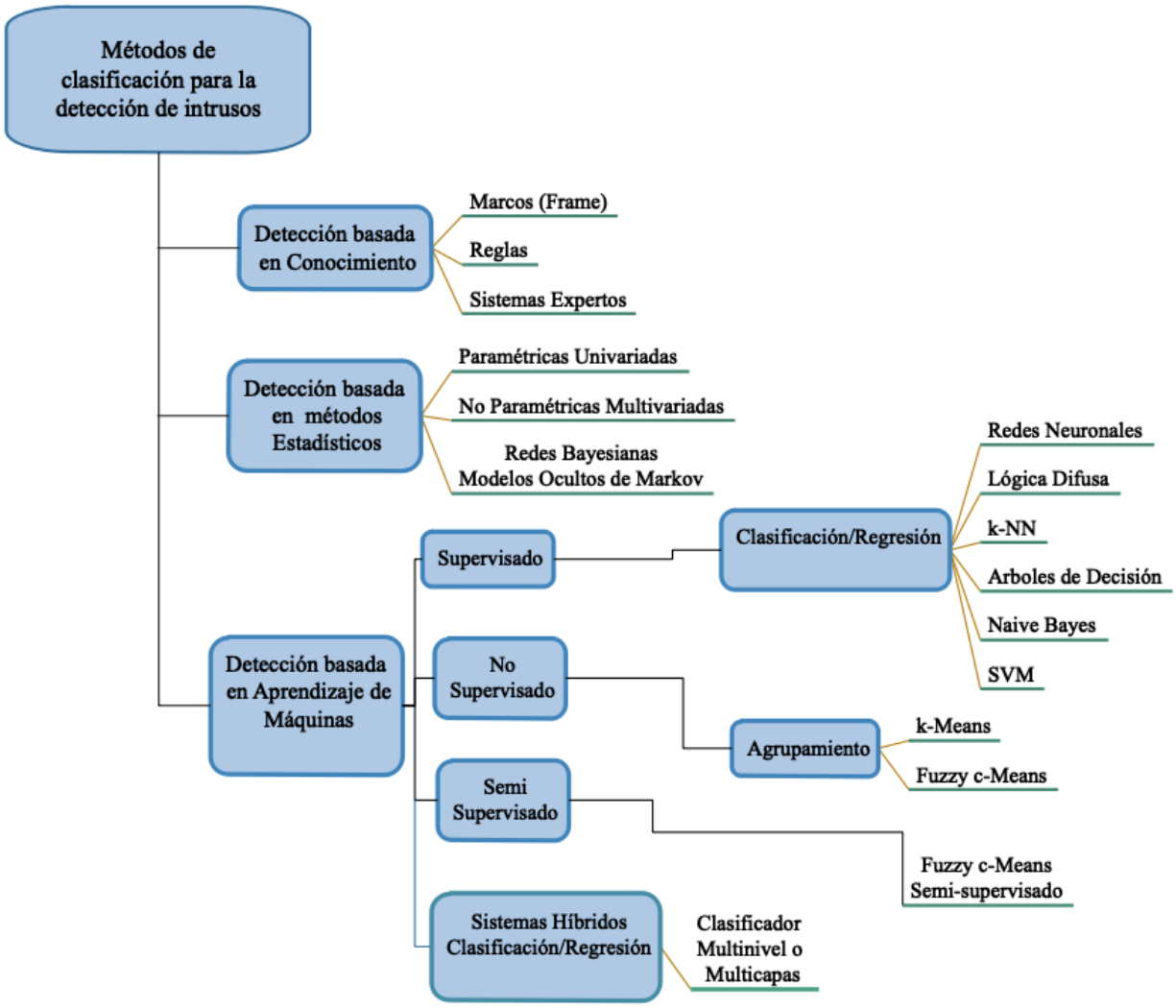


Figura 2-1.: Mapa conceptual del estado del arte

2.1. Métodos de Detección basados en Conocimiento

La información basada en el conocimiento, contiene las representaciones simbólicas de las reglas de juicio de los expertos, en un formato que le permite al motor de inferencia realizar una deducción sobre ella [21]. Así los IDS basados en el conocimiento propuesto en 1989 [22], depende de una base de datos de ataques conocida para determinar la actividad maliciosa, mediante la disección de la secuencia de bytes del tráfico de la red [23]. Tanto los sistemas de detección de intrusiones basados en firmas, como los sistemas de detección de intrusiones basados en anomalías, pueden utilizar el enfoque de detección basado en el conocimiento, al definir un conjunto de reglas, firmas o conocimientos predefinidos, como la base para decidir si un patrón es un intruso [24]. Aunque son sistemas con altos niveles de precisión en la

detección y un número mínimo de falsos positivos, incluso en intrusiones muy difíciles de detectar, una pequeña variación o actualización en los ataques conocidos, no serán detectados por el sistema, tal es el caso del Ataque del Día Cero, para el cual este tipo de sistemas es inviable [25].

Las técnicas basadas en Conocimiento se categorizan en modelos basados en marcos, reglas y sistemas expertos [26]. El modelo basado en marcos localiza un cuerpo completo de conocimientos y acciones esperadas en una única estructura, el basado en reglas es la forma modificada de estas con base en la gramática y los sistemas expertos tienen la intención de clasificar los datos de auditoría de acuerdo con un conjunto de reglas [21].

Un marco contextual de varios modelos de predicción de ataques en conjunto con IDS basados en conocimiento, es desarrollado en [27] para la exploración de ciberataques. El marco funciona sobre un top de IDS basados en reglas como Snort para mejorar su efectividad en la detección, para ello utilizan medidas de similaridad e inferencia semántica entre los ciberataques existentes y así crea relaciones contextuales entre ellos. Los ataques son representados como nodos en Redes de Enlace Semántico (SLNs por sus siglas en inglés). Los modelos de predicción creados se utilizan por capas en tiempo real para realizar tareas de expandir y filtrar las predicciones de los IDS, mejorando la tasa de detección de ataques. Si bien el enfoque identifica las relaciones entre los ataques no especifica el orden de su aparición. En [28] se presenta un sistema de detección de intrusiones de flujo, basado en firmas para redes de alta velocidad que se ejecutan en hardware básico. El sistema toma como entrada, las firmas HTTP de las reglas de Snort y las aplica a los elementos de información relacionados con HTTP en los flujos de IPFIX (Protocolo de Internet de Flujo de Información de Exportación por sus siglas en inglés). El algoritmo es denominado FIXIDS, detecta todos los eventos relevantes y presenta un rendimiento mayor que Snort con los mismos datos de red y el mismo conjunto de reglas, aunque no acepta las características estándar de flujo en las reglas.

Sistemas expertos para la detección de anomalías en sensores de datos inalámbricos [29], monitoreo y vigilancia del tráfico en carretera con sistemas de visión por computador [30], monitoreo y operación del sistema espacial [31] utilizan un conjunto de reglas para codificar en un motor de inferencia, la experticia del conocimiento humano y así tomar decisiones respecto a la seguridad del sistema. Así mismo en [32] realizan la detección de intrusos en dispositivos móviles, utilizando metodología de la Abstracción Temporal Basada en el Conocimiento (KBTA por sus siglas en inglés).

2.2. Detección basada en Métodos Estadísticos

Los métodos estadísticos para la detección de anomalías, utilizan el análisis estadístico para determinar el comportamiento del usuario y generar perfiles para representar su compor-

tamiento [33]. Se crean dos parámetros que son medidos para la detección, perfil actual y observado, los cuales incluyen diferentes medidas: Medida de la Intensidad de la Actividad, Medida de Distribución del Registro de Auditoría, Medidas Categóricas (distribución de una actividad en categorías) y Medida Ordinal (como el uso de la CPU). Un mecanismo de puntuación es utilizado para puntuar una actividad anómala, cuando la puntuación calculada excede cierto valor del umbral, se generará una alarma [23, 34]. De este modo, el comportamiento malicioso se diferencia del comportamiento normal, mediante el uso de propiedades estadísticas como la media y la varianza de las actividades normales y las pruebas estadísticas que determinan la desviación de las actividades del comportamiento normal, permitiendo encontrar valores de umbral precisos y disminuir la tasa de falsas alarmas. [33].

La literatura reporta técnicas paramétricas (modelos univariados) y no paramétricas (modelos multivariados), para desarrollar modelos estadísticos de detección de anomalías [11]. La primera supone que los datos normales se crean por parámetros y la puntuación de anomalías de una instancia de datos es la función de densidad de probabilidad (modelo Gaussiano o modelo de Regresión) relacionada con los parámetros, definiendo de esta manera un rango aceptable de valores para cada variable. En las técnicas no paramétricas la estructura del modelo a partir de los datos dados (basado en histogramas) y se considera la correlación entre dos o más variables. Estas técnicas hacen menos suposiciones con respecto a los datos, como la suavidad de la densidad [35]. Un método para modelar actividades complejas y detectar anomalías mediante el uso de modelos no paramétricos de los Procesos de Gaussianos (GP, por sus siglas en inglés) en una escena de tráfico concurrida, es propuesta en [36], donde los modelos GP aprenden la relación espacio temporal entre los patrones de actividad regional y predicen su distribución. Las anomalías son detectadas al comparar la predicción con la observación real. Jingjing Fei y Shiliang Sun [37] mejoran los métodos de detección de anomalías online existentes basados en GP, desarrollando el método de Procesos Gaussianos Dispersos con función Q (SGP-Q, por sus siglas en inglés), donde SGP-Q utiliza SGP con menor complejidad de tiempo para modelar datos de series de tiempo y acelera la detección de anomalías en línea.

Otras técnicas relevantes con base en modelos estadísticos para la detección de anomalías son los Modelos Ocultos de Markov (HMM, por sus siglas en inglés) y los estimadores de Bayes (Redes Bayesianas) [38,39]. En [40] se desarrolla un método de clasificación basado en el estado del Modelo oculto de Markov, para detectar ataques avanzados con una secuencia de etapas de ataque. El sistema correlaciona eficientemente los registros, reduce los falsos positivos y mejora la eficiencia del trabajo de administración de seguridad. La viabilidad de utilizar HMM para predecir ataques multipaso en tiempo real es demostrado en [41]. Este sistema puede proporcionar una detección temprana de ataques, previendo los pasos de los atacantes. De otra forma, en [42] es propuesto un marco de detección de intrusos con base en Redes Bayesianas usando enfoque Wrapper para la selección de características. El

enfoque reconstruye la base de datos NSL-KDD reduciendo las características de 41 a 16, en la clasificación de instancias se obtiene una tasa de falsos positivos del 0.7 % y 98,26 % para la tasa de verdaderos positivos.

2.3. Métodos Aprendizaje de Máquina

Las técnicas Aprendizaje de Máquina (ML, por sus siglas en inglés) permiten programar una máquina para realizar tareas, predicción, clasificación y/o agrupamiento de instancias por sí mismas o de forma automática, en presencia o ausencia de datos de entrenamiento. De este modo el Aprendizaje Automático basado en IDS proporciona una metodología de aprendizaje para clasificar y/o agrupar clases de ataques con base en el comportamiento normal y del ataque aprendido [7]. En los últimos años los ML-IDS han reportado algoritmos novedosos con alta precisión y mejor detección de ataques [43].

La literatura reporta algunas variaciones sobre cómo definir los tipos de algoritmos de Aprendizaje Automático, pero comúnmente se pueden dividir en:

2.3.1. Aprendizaje Supervisado

Los algoritmos para la detección de anomalías supervisado intentan modelar relaciones y dependencias entre la salida de predicción objetivo y las características de entrada, de manera que se puedan predecir los valores de salida para los nuevos datos, en función de las relaciones que aprendió de los conjuntos de datos anteriores. Los principales tipos de problemas de Aprendizaje Supervisado incluyen problemas de regresión y clasificación y las técnicas de ML de uso común en el campo de la detección de intrusos son Árboles de Decisión, Naive Bayes, Redes Neuronales Artificiales, Lógica Difusa, Máquinas de Vectores de Soporte y k-Vecinos más Cercanos (DT, NB, ANN, FL, SVM y k-NN, respectivamente por sus siglas en inglés) [44].

2.3.1.1. Árboles de Decisión

Un árbol de decisión se asemeja a la estructura de un árbol con hojas y ramas que pueden representarse como un conjunto de reglas if-then. [13]. El proceso de clasificación de una muestra se realiza pasando por una serie de decisiones, la primera decisión ayuda a la segunda y se convierte en una estructura de árbol. La clasificación de la muestra comienza con el nodo raíz y termina con el nodo final, que también se denomina nodo hoja. Cada nodo final (nodo hoja) representa una categoría de clasificación [45]. La literatura reporta numerosos clasificadores basados en el árbol de decisión, aunque los más conocidos son ID3 [46] y C4.5 [47] con sus versiones mejorados por [48, 49].

[50], propuso Sistema eficiente de detección de intrusiones basado en host que utiliza un DT parcial y un algoritmo de selección de características de correlación. El enfoque propuesto extrae las características basadas en la correlación y usa el DT parcial para la clasificación. El conjunto de datos de KDD99 se utiliza para la evaluación del desempeño y los resultados obtenidos muestran que el CPDT propuesto supera los enfoques convencionales con una precisión del 99.9 %. De igual manera en [51] se implementó un Sistema de detección de intrusiones basado en el DT sobre Big Data en un entorno de niebla. El sistema es evaluado en el conjunto de datos KDD99 y comparado con múltiples métodos en conjunto de datos del 10 % y el conjunto de datos completo, los resultados mostraron efectividad en el método propuesto.

2.3.1.2. Redes Neuronales Artificiales

Las Redes Neuronales Artificiales son modelos computacionales que imitan la estructura neuronal del cerebro humano. Así (ANN, por sus siglas en inglés) es una red de capas de neuronas artificiales [44], agrupadas en una colección de elementos de procesamiento altamente interconectados para realizar una transformación de entrada-salida. La red neuronal gana conocimiento sobre la transformación determinada por el conjunto de pesos asociados con enlaces de elementos, mediante el aprendizaje iterativo de un conjunto de muestras de entrenamiento. Así, al modificar las conexiones entre los nodos, la red puede adaptarse a las salidas deseadas [52].

La literatura reporta diferentes investigaciones en IDS con base en ANN, en [53] utilizan la arquitectura Perceptron Multicapa para la clasificación de patrones, normales, ataques y sus tipos. En [54] presentan un sistema de reducción de características previo a la clasificación de datos de prueba, en clases de ataque y no ataque. El sistema de de selección se implementa utilizando ganancia de información y correlación, la etapa de clasificación es realizada con una Red Neuronal de Realimentación. Ambos métodos fueron validados con la base de datos KDD Cup 99. En [19] un enfoque de detección de intrusiones en el internet de las cosas (IoT, por sus siglas en inglés) es implementado para identificar ataques DDoS / DoS. El clasificador ANN fue validado en una red simulada de IoT que demuestra una precisión de más del 99 % de desempeño.

2.3.1.3. Lógica Difusa

El concepto de lógica difusa fue concebido por Lofti Zadeh [55] como una forma de procesar datos al permitir una membresia o grado de pertenencia de un dato a un conjunto de características similares, para modificar la clasificación exclusiva o dicotómica. Esté método proporciona una flexibilidad muy valiosa para el razonamiento que toma en cuenta las inexactitudes y las incertidumbres, como es el caso de los ataques en la red [56–58].

Un Multiclasicador basado en Lógica Difusa para la detección de nuevos atques DoS/DDoS es propuesto en [59]. Para detectar el comportamiento de la intrusión se utiliza un algoritmo de reglas de asociación incremental de una pasada, el método es implementado en la base de datos KDD Cup 99 obteniendo una precisión en la detección del 98.2 %. También se investigó en [60] sobre un sistema de detección de intrusiones de red basado en lógica difusa para predecir el ataque Neptuno, que es un tipo de ataque de inundación en el protocolo TCP. El método propuesto genera las reglas de decisión mediante una estrategia de permutación de características, lo que le permite al sistema obtener mejor precisión en presencia de cambios menores en los datos de entrenamiento, debido a la elasticidad de los conjuntos difusos. En [61] Diseñaron un módulo de toma de decisiones difusas para construir el sistema más preciso de detección de ataques, utilizando el enfoque de inferencia difusa. Un conjunto efectivo de reglas difusas para el enfoque de inferencia se identificó automáticamente haciendo uso de la estrategia de aprendizaje de reglas difusas.

2.3.1.4. Máquinas de Vectores de Soporte

Máquinas de Vectores de Soporte (SVM, por sus siglas en inglés) es un método ampliamente aplicado en los últimos años para proporcionar soluciones potenciales en la detección. la selección del kernel apropiado y sus parámetros de acuerdo al problema de clasificación influyen de forma significativa en el rendimiento de la SVM, determinando la capacidad de generalización y aprendizaje [62]. La literatura reporta el uso del kernel Gaussiano en la mayoría de los IDS dependiendo de la naturaleza del problema de clasificación [63–65].

En [66], construyeron un modelo de detección de intrusos utilizando SVM en la plataforma Big Data de Apache Spark, utilizaron KDD99 para entrenar y probar el modelo, los resultados del experimento mostraron que el modelo SparkChi-SVM tiene un alto rendimiento, reduce el tiempo de entrenamiento y es eficiente para Big Data. También en [67], implementaron la transformación de las proporciones de densidad marginal de logaritmo, para formar las características originales, con el objetivo de obtener características transformadas nuevas y de buena calidad que puedan mejorar considerablemente la capacidad de detección en SVM. El conjunto de datos NSL-KDD se usa para evaluar el método propuesto, y los resultados empíricos muestran que logra un rendimiento mejor y más robusto que los métodos existentes en términos de precisión, tasa de detección, tasa de falsas alarmas y velocidad de entrenamiento. Gu y otros [68] desarrollaron un enfoque novedoso para la detección de intrusiones utilizando SVM con aumento de características que mejora el rendimiento y robustez del método anterior al elegir una combinación no lineal que es agregada al clasificador SVM

2.3.1.5. k-Vecinos más Cercanos (k-NN)

k-Vecinos más Cercanos (k-NN, por sus siglas en inglés) es un método basado en instancias, supervisado, y fue propuesto por Fix y Hodges en 1951 [69]. Es uno de los algoritmos

más simples y precisos para la clasificación de patrones y modelos de regresión, a pesar de que su rendimiento compite con clasificadores más complejos de la literatura [18]. k-NN se desarrolló a partir de la necesidad de realizar análisis discriminantes cuando las estimaciones paramétricas confiables de densidades de probabilidad son desconocidas o difíciles de determinar [70] . Dentro de sus ventajas se encuentra la selección adecuada del parámetro k, responsable del tamaño del vecindario y la función de distancia, lo cual afecta la calidad de la representación topológica, que puede tener un impacto significativo en los resultados subyacentes, su efectividad para conjuntos de datos con un gran número de muestras y la garantía del error, cuya tasa de error asintótica es a lo sumo el doble de la tasa de error de Bayes [71].

Diferentes enfoques de k-NN y sus mejoras han sido propuestos en la literatura durante los últimos años, con el fin de abordar problemas de clasificación en IDS [6]. Métodos de detección con k-NN y medidas de similaridad [72], k-means para agrupamiento en el entrenamiento de los datos y k-NN para evaluar la calidad de éstos en un IDS [73] logran rendimiento satisfactorio en los conjuntos de datos de referencia de detección de intrusos probados en KDD Cup 99 y NSL-KDD. Por otro lado un método de programación genética con k-NN para la detección de intrusos es propuesto en [74], el modelo utiliza la base de datos KDD Cup 99 y realiza la selección de características óptimas con programación genética. Luego el proceso de clasificación entre normal y tipo de ataque es conducido con el método k-NN, donde se obtuvo una precisión de 99.60 % para todo el clasificador. En [75] propuso un IDS para una red de sensores inalámbricos utilizando el algoritmo de clasificación k-NN. El sistema detecta ataques de inundación de la red lo cual afecta seriamente el tráfico de datos, además en el trabajo de investigación se elabora un método de prevención de los ataques de inundación de manera más eficiente. Así mismo en [76] se implementaron algoritmos de clasificación usando k-NN.

2.3.1.6. Naïve Bayes

Es un clasificador basado en el modelo de probabilidad Bayesiano y opera con un fuerte supuesto de independencia, es decir, la probabilidad de un atributo no afecta la probabilidad del otro [77]. Aunque es un modelo reconocido desde los años 70, el clasificador *Naïve Bayes* aparece por primera vez en la literatura del aprendizaje automático a finales de los años 80, con el objetivo de comparar su capacidad predictiva con la de métodos más sofisticados. De manera gradual los investigadores de la comunidad científica en aprendizaje automático se han dado cuenta de su potencialidad y robustez en problemas de clasificación supervisada [78]. Gujar y Patil en [79] desarrollaron un IDS usando *Naïve Bayes* para datos en tiempo real. El sistema captura paquetes en tiempo real, con ellos conforma el conjunto de entrenamiento y usa el clasificador*Naïve Bayes* para detectar paquetes normales o intrusos. Un modelo de detección de intrusos mediante la selección de funciones de *Chi Square* y el clasificador

Naïve Bayes modificado, es propuesto en [80]. En este, un subconjunto de datos óptimo es encontrado utilizando Análisis Discriminante Lineal y la estadística de Chi Cuadrado para obtener los atributos de valor de prueba máximo. Finalmente el subconjunto es utilizado por el clasificador para identificar los ataques y conexiones normales. La base de datos NSL-KDD es utilizada para la experimentación demostrando que los ataques mayoritarios pueden clasificarse con precisión, mientras los ataques minoritarios y las conexiones de normal obtuvieron precisiones menores debido a la poca cantidad de muestras de entrenamiento que ilustran su comportamiento. En [81] propusieron el modelo OCPAD, donde adaptaron un clasificador *Naïve Bayes* Multinomial de una clase para detectar tipos de ataques en el protocolo HTTP. OCPAD un IDS de anomalías de contenido para detectar cargas útiles de paquetes anómalas en el tráfico de red. El experimento se llevó a cabo con un millón de paquetes HTTP recopilados en una red académica, obteniendo una tasa aceptable de detección de falsos positivos, menos del 6 %.

2.3.2. Aprendizaje no supervisado

A diferencia del aprendizaje supervisado, en el aprendizaje no supervisado solo se otorgan las características, sin proporcionarle al algoritmo ninguna etiqueta, es decir no es necesario disponer de la respuesta correcta en los datos de entrenamiento, ya que no se busca la reproducción de un resultado conocido, sino el descubrimiento de nuevos patrones o resultados [82]. Por tanto, su tarea es descubrir los patrones o estructuras ocultos de los datos en los que no existe una variable objetivo para realizar los métodos de clasificación o de regresión [12].

La reducción del conjunto de características originales a un conjunto más pequeño y más manejable, donde se pueden encontrar estos patrones se realiza agrupando instancias similares de datos. Esto se conoce como agrupación en clústeres y se puede lograr con una variedad de algoritmos de aprendizaje no supervisados, para usar en diferentes aplicaciones del mundo real [83]. En general, hay dos propósitos para usar análisis de *cluster*: comprensión y utilidad. El *cluster* por comprensión utiliza el análisis de agrupación para encontrar automáticamente grupos de objetos conceptualmente significativos, que comparten características comunes. Mientras la agrupación por utilidad permite abstraer los prototipos o los objetos representativos de los individuales, en las mismas agrupaciones. Estos objetos y prototipos sirven como base de varias técnicas de procesamiento de datos [84].

2.3.2.1. k-Means

El algoritmo *k-means* también llamado *k-means clustering* es uno de los algoritmos de *clustering* más antiguos y aún más utilizados, de hecho, ha sido identificado como uno de los 10 mejores algoritmos en métodos de Aprendizaje de Máquina [85]. En este método k indica el número de *cluster* que debe ser predefinido inicialmente, para luego realizar la selección

de los centroides iniciales de forma aleatoria y finalmente implementar un proceso iterativo de asignación de cada punto de datos a su centroide más cercano, donde cada colección de puntos asignados a un centroide forma un grupo. El centroide de cada grupo se actualiza según los puntos asignados a ese grupo. Este proceso es repetitivo hasta cumplir los criterios de convergencia [14]. En conclusión, *k-means* es un algoritmo de agrupación de particiones simple y basado en un prototipo que intenta encontrar k agrupaciones no superpuestas. Estos *cluster* están representados por sus centroides (un centroide de *cluster* suele ser la media de los puntos en ese *cluster*) [84].

Considerando numerosos algoritmos de agrupación propuestos en la literatura [86, 87], este estudio se enfoca en algoritmo *k-means clustering* aplicados a la detección de intrusos utilizando la base de datos KDD Cup 99. En [88] Propusieron un método que integra *k-means clustering* en el Algoritmo de Células Dendríticas (DCA, por sus siglas en inlés) para mapear los valores de contexto con base en la dinámica de la migración de células semi-maduras y maduras, para la formación de cluster semi-maduros (normales) y maduros (ataques), y de esta manera mejorar la precisión de clasificación. Los resultados experimentales arrojaron una precisión del 98.01 % en la clasificación. Sukumar y otros [89] desarrollaron un sistema de detección de intrusiones que utiliza un Algoritmo Genético *k-means* (IGKM por sus siglas en inglés) mejorado para detectar el tipo de intrusión, el sistema usa la agrupación en *cluster* para detectar el tipo de spam. IGKM utiliza una función de aptitud que ayuda a encontrar el valor óptimo de k con alta precisión. Así mismo, [90] propuso utilizar la clasificación optimizada de mínimos cuadrados regularizados en combinación con *k-means*. El método usa *k-means* para obtener vectores base optimizados adecuados para la clasificación de cada tipo de ataque.

2.3.2.2. *Fuzzy c-Means*

El concepto de conjuntos difusos fue desarrollado por Zadeh en 1965 como un intento de modificar la agrupación exclusiva o dicotómica en función de su probabilidad, considerando cualquier parámetro en el que se realizaron las agrupaciones [55]. El concepto es ampliado a *Fuzzy c-Means* (FCM, por sus siglas en inglés) por Bezdek y otros [91] como un algoritmo de *cluster* para mitigar la capacidad de agrupación de datos inciertos, vagos y difíciles de agrupar, asignado grados de pertenencia entre 0 y 1 mediante el uso de una matriz de membresía [92]. El uso de esta matriz incrementa la expresividad del análisis de agrupamiento, posibilitando una vista más completa de las relaciones presentes en los datos [93].

FCM ha sido un método clásico de agrupación en clústeres para el análisis de datos en el campo de Aprendizaje de Máquina, aplicado con éxito en la detección de intrusos en la red, como se reporta en la literatura [94–99]. Los modelos utilizan FCM para realizar las particiones difusas y la agrupación de datos, con el objetivo de mejorar la sensibilidad a

los valores iniciales, la convergencia del centro del *cluster* de FCM y obtener una mayor capacidad discriminante entre los datos normales y ataques. Los sistemas son validados mediante la base de datos estándar KDD Cup 99 y los resultados experimentales muestran la efectividad de éstos para detectar ataques y reconocer el comportamiento normal con altas tasas de detección, incluso para ataques repetidos y bajas tasas de detección para falsas alarmas. Finalmente, en [15] presentan un método IDS denominado Optimización de Arroz Híbrido (HRO por sus siglas en inglés) mejorado con FCM, el cual permite mejorar o acelerar la velocidad de convergencia de algoritmo FCM. Al igual que en los métodos anteriores el experimento es implementado en la base de datos KDD Cup 99.

2.3.3. Semi Supervisado

El aprendizaje semi supervisado es una combinación de métodos de aprendizaje supervisados y no supervisados. Se ocupa de las tareas de aprendizaje utilizando datos etiquetados y no etiquetados [100]. Tiene como objetivo aprovechar los datos no etiquetados para mejorar el rendimiento [101]. Un gran número de algoritmos de aprendizaje semi supervisados optimizan conjuntamente dos funciones de objetivo de entrenamiento: la pérdida supervisada sobre los datos etiquetados y la pérdida no supervisada sobre los datos etiquetados y no etiquetados. Este método ha atraído mucha atención de la comunidad científica.

2.3.4. Clasificador Multinivel o Multicapas

La finalidad de este modelo, es utilizar un conjunto de varias técnicas de aprendizaje de máquina, donde se discuten las técnicas de análisis supervisado, no supervisado y semi supervisado como: SVM, KNN, ANN, Lógica Difusa, Fuzzy c-Means, Arboles de Decisión, entre otras, donde se puedan considerar posibles clasificaciones que se realizan de manera individual y combinarlo para obtener una clasificación global que supere en rendimiento la intervención de cada método por separado [102]. La integración de varios niveles en los clasificadores mejoran el rendimiento de las técnicas de detección de intrusos, especialmente para la detección de ataques de baja continuidad [103]. Para este tipo de configuración se han realizado múltiples investigaciones: [104], que Implementaron un clasificador híbrido multinivel que usa diferentes conjuntos de características sobre cada clasificador, para obtener mayor rendimiento para identificar tasas de falsos positivos y mejor desempeño global en identificar tráfico de red anormal como ataques. En este contexto para [100] Utilizaron la red neuronal con pesos aleatorios (NNRw, por sus siglas en inglés) como un clasificador base, debido a que es computacionalmente más eficiente y tiene un excelente rendimiento de aprendizaje. Los parámetros del nodo oculto (es decir, pesos y datos sesgados) en NNRw se seleccionan de forma aleatoria e independiente para obtener mejor precisión en el clasificador final. En [105] propusieron un método de selección de características de dos niveles basado en mRMR por sus siglas en inglés y ganancia de información para LDRA por sus siglas en

inglés en la detección de intrusos. El método utiliza mRMR para filtrar características irrelevantes, reduce la dimensión de los datos y mejora las tasas de detección de algunas clases mediante el cálculo de ganancia de información, para luego combinar estas características juntas. Finalmente, se comprueba la validez y precisión del método.

3. Marco Teórico

Las redes de computadoras han evolucionado en los últimos años para permitir un intercambio de información sin precedentes entre individuos, corporaciones e incluso entidades gubernamentales [106]. De la misma forma, la necesidad de proteger esta información es un reto en constante cambio, y la seguridad de la red se ha convertido en una preocupación esencial no sólo en la comunidad científica, sino a nivel empresarial y gubernamental. Incluso en organizaciones más pequeñas, el objetivo básico de prevenir el acceso no autorizado y al mismo tiempo permitir que la información legítima fluya sin problemas requiere el uso de Sistemas de Detección de Intrusos [107].

3.1. Redes Computacionales

En términos generales, una red computacional es una colección de individuos (nodos) interconectados entre sí (relaciones implícitas o explícitas) para compartir recursos con alta confiabilidad y acceso flexible. Dichas relaciones pueden ser estrictamente físicas o conceptuales, como alguna similitud entre pares o dentro de un par. Así mismo, en una red implícita o redes de afinidad, los individuos desconocen sus relaciones, mientras en una red explícita los individuos están familiarizados con al menos sus vecinos locales y dan cuenta de la conexión proyectada, es decir su similitud [108].

Una red computacional se puede representar matemáticamente por $G = (V, E)$ [109], donde V es el conjunto de nodos y E es el conjunto de líneas o interconexiones. El número de nodos usualmente es denotado por $n = |V|$ y el número de conexiones es $m = |E|$. En este contexto la red se representa como una matriz de adyacencia A, donde cada elemento a_{ij} es denotado como:

$$\begin{cases} w_{ij} & if(i,j) \in E \\ 0 & if(i,j) \notin E \end{cases} \tag{3-1}$$

Donde w_{ij} es el peso de las conexiones entre los nodos i y j. Cuando $(i, j) \in V$ significa una conexión entre los nodos i y j.

En una red típica de computadores (ver Figura **3-1**), hay normalmente tres entidades que se comunican: los usuarios o personas que realizan diversas actividades en la red ya sea para compartir recursos o servicios, los hosts que son dispositivos identificados con una dirección única IP y los procesos o actividades de programas ejecutables en una arquitectura

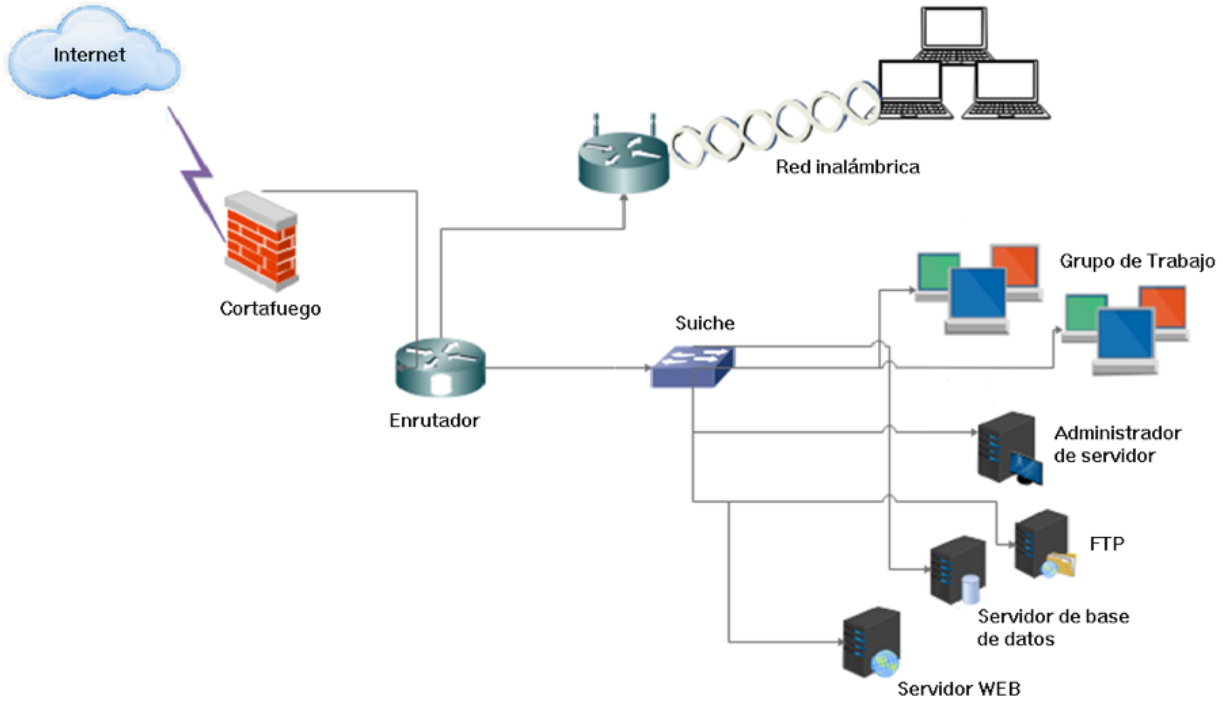


Figura 3-1.: Red típica de computadores

cliente-servidor [110]. Por otro lado el rendimiento del intercambio de datos entre los nodos depende en gran medida del tipo de medio de comunicación (ver Figura **3-2**), así como de los dispositivos utilizados, por ello hoy en día es evidente los avances en los medios de comunicación en cuanto a velocidad, fiabilidad, robustez y costo [111, 112].
Los tipos de redes y sus diseños se pueden clasificar según su topología, la escala geográfica y la forma de establecer la comunicación. Así, la topología puede ser física o lógica. Las primeras son la representación geométrica de la forma de enlace entre los dispositivos, mientras la segunda representa la forma del flujo de los datos. Los tipos de topologías más comunes son: Bus, Anillo, Árbol y Estrella. En cuanto a la extensión geográfica de la red se disponen de las Redes de Área Local, Redes de Área Metropolitana y Redes de Área Extendida (LAN, MAN y WAN respectivamente por sus siglas en inglés). Finalmente, las redes WAN proporcionan medios de transmisión a largas distancias en centenares o miles de kilómetros y la forma en que establecen la comunicación se denomina conmutación de circuitos y conmutación de paquetes [113, 114].

Desde otra perspectiva la arquitectura de red se puede precisar como el conjunto de capas y protocolos que integran un sistema de comunicaciones. Cada capa o nivel es un consumidor de servicios ofrecidos por el nivel inferior y proveedor de servicios por el nivel superior.

La comunicación entre entidades de una misma capa, en distintos dispositivos es adminis-

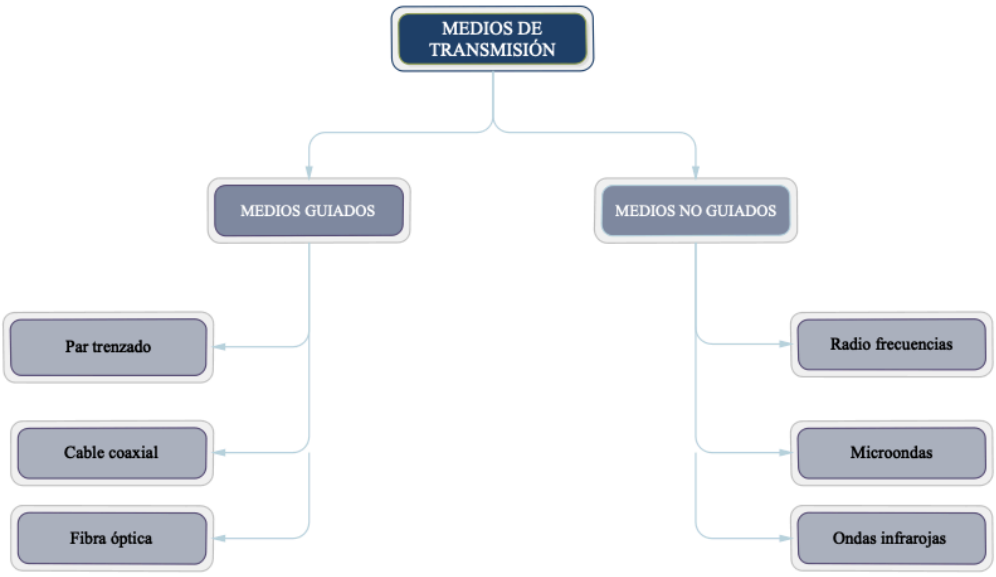


Figura 3-2.: Medios de transmisión Fuente:[113]

trada por un conjunto de reglas denominadas protocolos, también cuando la comunicación se produce entre entidades de capas diferentes de un mismo dispositivo. Al conjunto de reglas que administra dicho intercambio de información se le denomina interfaz. Las dos arquitecturas de red fundamentales para el desarrollo de estándares de comunicaciones se denominan: modelo OSI y modelo TCP-IP, como se describe en la (Figura **3-3**) [113, 115].

Existen tres elementos fundamentales para la seguridad de las redes computacionales que son: *1)* la confidencialidad, donde brinda la garantía de acceder a la información de los usuarios que se encuentran autorizados para tal fin; *2)* la integridad, que abarca la preservación de la información completa y exacta; *3)* la disponibilidad, que debe garantizar que el usuario acceda a la información que necesita en el momento oportuno [34].

3.2. Ataques a las Redes Computacionales

Son acciones que se toman para exponer, alterar, interrumpir, destruir, robar y obtener acceso sin autorización a los activos y la información de las operaciones normales en una red, al explotar vulnerabilidades, utilizando diversas técnicas y herramientas. Estas amenazas pueden ser: internas y externas. Los ataques internos no siempre son llevados a cabo por el personal de la organización que intenta escalar privilegios sin autorización, o usuarios que ya tienen estos privilegios para cometer algún delito informático, acceder a la información, indisponer el servicio y modificar la información de forma mal intencionada; mientras que los ataques externos, son usuarios que están en otras redes tratando de vulnerar la red con

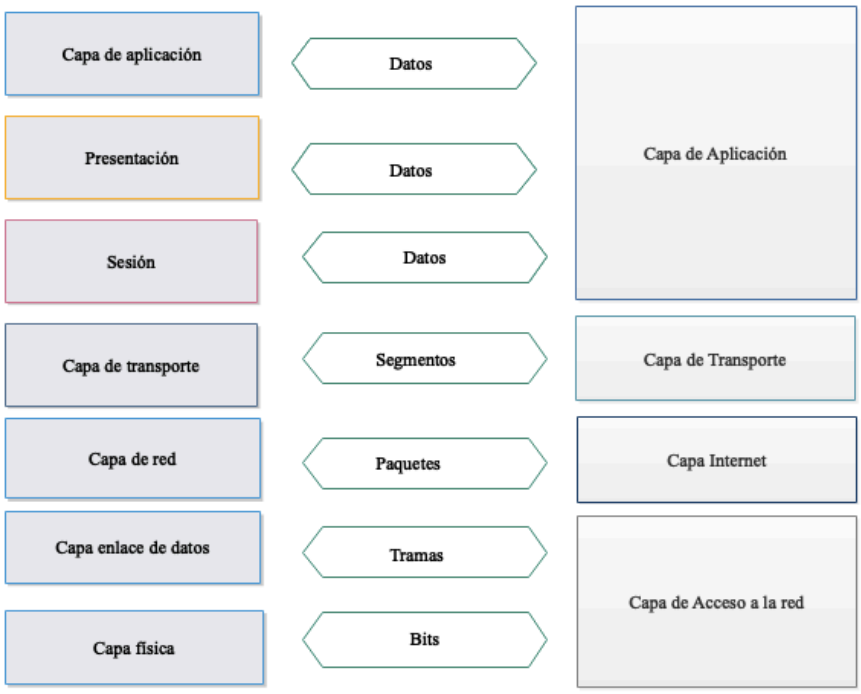


Figura 3-3.: Modelo OSI Modelo TCP/IP Fuente:[116]

diferentes tipos de técnicas. Estos tipos de ataques se clasifican en activos y pasivos, donde los primeros, intentan alterar los recursos del sistema o afectar sus operaciones. El ataque activo implica alguna interrupción, modificación y fabricación de la secuencia de datos o la creación de una declaración falsa. El segundo, intenta aprender o hacer uso de la información del sistema, pero sin afectar los recursos de la red. Éstos ataques se dan en forma de escucha y monitoreo de la transmisión que pasa por la red. La finalidad de estos tipos de ataques es obtener información que se está transmitiendo [116]. Estos tipos de ataques se listan en la Tabla **3-1**.

Tabla 3-1.: Clasificación de los tipos de ataques.

Activos	**Pasivos**
Enmascaramiento	Obtención de la información
Retransmisión	Análisis de tráfico
Modificación de mensajes	Divulgación del contenido
Denegación de servicios	Obtención de los parámetros de origen y destino de la comunicación

3.3. Sistema de Detección de Intrusos (IDS)

El sistema de detección de intrusos monitorea el tráfico de la red para analizar patrones de intrusiones mediante la recopilación de datos de diferentes sistemas y fuentes de la red, al igual que la comprobación de los datos de posibles amenazas. En este contexto, la función del IDS es alertar sobre posible trafico malicioso para así tomar las medidas correctivas cuando son detectadas [19]. Los IDS se clasifican en: sistemas basado en Host, los cuales son dispositivos que se instalan en una máquina especifica, para examinar el estado de la información sobre el comportamiento del sistema que alerta sobre posibles anomalías. Por otro lado se encuentran los IDS basados en red, por lo tanto son sistemas que analizan eventos relacionados con la red, como el volumen de tráfico, la dirección IP, los puertos de servicio y el protocolo utilizado [117–119].

3.3.1. Tipos de sistemas de detección de intrusos

Existen diferentes enfoques para detectar actividades maliciosas y de irrupción mediante el sistema de detección de intrusión basado en la infraestructura del sistema y el tipo de detección de intrusión. Donde se encuentran dos grandes métodos, como lo son: la detección basada en anomalías y la detección basada en usos indebido o firmas [120].

- **Detección de anomalías:**

  Se refiere a la identificación de eventos que parecen ser sospechosos o irregulares con respecto al comportamiento normal del sistema, donde se identifican actividades que no son usuales para los patrones predeterminados de esta estructura [121, 122]. La detección de anomalías usualmente implica la creación de bases de conocimiento que contienen los perfiles de las actividades, donde generalmente se utiliza el monitoreo de umbrales para indicar cuándo se ha alcanzado una determinada métrica establecida, donde su principal beneficio es el alcance de detección de nuevos ataques [34].

- **Detección de usos indebidos:** La detección de uso indebido se basa en el conocimiento de las vulnerabilidades del sistema y los patrones de ataque conocidos. Este tipo de técnica se ocupa de encontrar intrusos que intenten ingresar a un sistema mediante la explotación de una vulnerabilidad conocida. El escenario de intrusión, es una serie de eventos que pueden resultar en una intrusión sin alguna intervención preventiva externa. Este sistema de detección de intrusos compara continuamente las actividades recientes para encontrar escenarios de intrusión conocidos. Es necesario asegurar que uno o más atacantes no estén intentando explotar vulnerabilidades conocidas. Para realizar esto, cada escenario de intrusión debe ser descrito o modelado [34].

3.4. Selección de características

La selección de características (SC) es usada para identificar y remover las variables que no contribuyen a los procesos de regresión y clasificación (información redundante e irrelevante), con el objetivo de filtrar las principales características, mejorar el rendimiento y precisión de los predictores y/o clasificadores, así como reducir el costo computacional. Este proceso es descrito como la obtención de un subconjunto de m características a partir de un conjunto original de n características, donde $m < n$, de modo que se conservan las características relevantes o que aportan información del sistema y se eliminan las redundantes e irrelevantes. La estimación se obtiene mediante una función de evaluación, que se compara constantemente con el anterior, hasta obtener el mejor de ellos por medio del criterio de parada. Mediante la validación se verifica la calidad que satisface las condiciones del proceso [123, 124].

3.4.1. Conjuntos Rough

Está basada en aproximaciones, parte de la suposición de que cada objeto (evento) del universo se asocia con alguna información (datos o conocimiento). Generalmente existen eventos que se caracterizan por tener la misma información, haciéndolos imperceptibles al conocimiento disponible acerca de ellos, razón por la cual la estadística clásica o modelos de análisis convencionales no tienen la capacidad de discernir, limitando su conocimiento acerca de los elementos del universo [125]. El método de conjuntos rough surge como una solución a problemas tales como la exploración de patrones ocultos, reducción de datos y análisis de relevancia, permitiendo explorar el conocimiento desde las bases de datos por medio de una matriz de discernibilidad, que a su vez identifica como las características de un evento que difieren entre si antes de la clasificación. El reconocimiento de la teoría de conjuntos rough ha sido de gran importancia para las investigaciones y desarrollo de máquinas de aprendizaje, dando lugar a varias extensiones de la teoría original y ampliando cada vez más su campo de aplicación [123, 125].

3.4.2. Conjuntos Fuzzy

La teoría de la lógica difusa surge como una metodología para la formulación y solución de problemas que no pueden ser definidos dentro de la lógica convencional de Boole, por ejemplo, la descripción matemática de la ambigüedad y la ambivalencia [125]. Un conjunto difuso puede ser definido como un conjunto de pares ordenado $A = \{x, \mu_A(x) | x \in \mathbb{U}\}$. La función $\mu_A(x)$ llamada función de membresía es el grado de pertenencia de x a $\mathbb{U}$, de modo que asigna cada elemento del universo a un grado de membresía en el rango $[0, 1]$. Cuanto más cerca esté A del valor 1, mayor será la pertenencia del objeto x al conjunto A. Así, los valores de pertenencia varían entre 0 cuando no pertenece en absoluto y 1 cuando hay pertenencia total. El universo puede ser continuo o discreto. Cualquier conjunto difuso que contenga al menos un elemento con un grado de membresía de 1 es llamado normal.

3.4.3. Fuzzy Rough Sets

Fuzzy Rough Sets (FRS) encapsula los conceptos relacionados pero distintos de la vaguedad por conjuntos difusos y la indiscernibilidad por conjuntos rough, los cuales se producen como consecuencia de la incertidumbre en el conocimiento. La selección de características con FRS proporciona un medio por el cual datos ruidosos de valor real o discreto (o la mezcla de ambos) pueden ser efectivamente reducidas sin necesidad de información proporcionada por el usuario. Además esta técnica puede ser aplicada a datos con atributos de decisión continuo o nominal [125].

Sea un conjunto no vacío de objetos finitos (el universo del discurso) donde $x, y \in \mathbb{U}$ y $\mathbb{A}$ es un conjunto no vacío finito de características donde a es una característica en $\mathbb{A}, p \subseteq \mathbb{A}; y$ $\mathbb{D}$ y es un conjunto de características de decisión. Las aproximaciones altas y bajas difusas pueden ser definidas usando una relación transitiva de similaridad difusa Υ para aproximar una clase de equivalencia difusa X [123]:

$$\mu_{\underline{Rp}X}(x) = \inf_y \Psi\left(\mu_{Rp}(x,y), \mu_x(y)\right) \tag{3-2}$$

$$\mu_{\overline{Rp}X}(x) = \sup_y \Upsilon\left(\mu_{Rp}(x,y), \mu_x(y)\right) \tag{3-3}$$

Donde Ψ es una inferencia difusa y Υ una intersección difusa. R_p es la relación de similaridad difusa inducida por el subconjunto de características P:

$$\mu_{Rp}(\chi, y) = \Upsilon_{a \in P}\{\mu_{Ra}(\chi, y)\} \tag{3-4}$$

μ_{Ra} es el grado en que los objetos x y y son similares para la característica a. La región positiva clásica en la teoría de conjuntos rough tradicional es definida como la unión de las aproximaciones bajas. Por el principio de extensión [126] la función de membresía de un objeto $x \in \mathbb{U}$ perteneciente a la región positiva difusa puede ser definida por [127]:

$$\mu_{POSp\mathbb{D}^{(\chi)}} = \sup_{x \in \frac{\mathbb{U}}{\mathbb{D}}} \mu_{\underline{Rp}x}(\chi) \tag{3-5}$$

Una cuestión importante en el análisis de datos es el descubrimiento de la dependencia entre los atributos. El grado de dependencia de $\mathbb{U}$ del conjunto fuzzy rough en el subconjunto de atributos P puede ser definido:

$$\gamma'_p\mathbb{D} = \frac{\sum_x \mu_{POSp((D)^{(\chi)}}}{|\mathbb{U}|} \tag{3-6}$$

Un reducto $\mathbb{R}$ del conjunto fuzzy rough puede ser definido como un subconjunto mínimo de características del conjunto de atributos inicial $\mathbb{C}$ tal que para un conjunto dado de atributos $\mathbb{D}$ conserva el grado de dependencia del conjunto de datos, es decir, $\gamma'_R\mathbb{D} = \gamma'_\mathbb{C}\mathbb{D}$ y $\gamma'_{R-(a)}\mathbb{D} \neq \gamma'_R\mathbb{D}$ para todo $a \in \mathbb{R}$.

En el **Algoritmo 1**, se presenta una rutina conocida como QUICK-REDUCT frecuentemente usada en la literatura.

Algoritmo 1: Algoritmo QUICK-REDUCT

Requiere: $\mathbb{C}$, el conjunto de todas las características condicionales;
$\mathbb{D}$, el conjunto de todas las características de desición.
$R \leftarrow \{\}\,; \gamma'_{best} = 0; \gamma'_{prev} \neq 0$
while$; \gamma'_{best} = 0; \gamma'_{prev}$ **do**
  $\mathbf{T} \leftarrow \mathbf{R}$
  $\gamma'_{best} = \gamma'_{prev}$
  for all$x \in (C - R)$ **do**
    if$\gamma'_{RU\{x\}}\mathbb{D} > \gamma'_T\mathbb{D}$ **then**
      $\mathbf{T} \leftarrow \mathbf{R} \cup \{x\}$
      $\gamma'_{best} = \gamma'_T\mathbb{D}$
    end if
  end for
  $\mathbf{R} \leftarrow \mathbf{T}$
  end while
  return R
  Salida: R, subconjunto mínimo de características

3.4.4. *Relief F*

Relief F fue un algoritmo propuesto por Kononenko en 1994 [128], el cual es una extensión del algoritmo Relief [129], que mejora los problemas asociados a la clase binaria, permitiendo trabajar con conjuntos de datos multiclases, ruidosos e incompletos. Ambos métodos son algoritmos de ponderación de funciones supervisados que utilizan el modelo de filtro. *Relief F* evalúa la relevancia de las características por su capacidad para discernir instancias de una clase a otra en una vecindad local, es decir, las características seleccionadas como mejores son aquellas que más contribuyen a aumentar la distancia entre las diferentes instancias de la clase y disminuyen la distancia entre instancias de la misma clase [130]. El criterio de evaluación de *Relief F* es definido como [131]:

$$\varphi_R(F_i) = \frac{1}{M} \cdot \sum_{t=1}^{M} \left\{ -\frac{1}{M_{t,CL(\mathbf{x}_t)}} \sum_{\mathbf{x}_j \in NH(\mathbf{x}_t)} \|x_{t,i} - x_{j,i}\| \right. \\ \left. + \sum_{C \neq CL(\mathbf{x}_t)} \left(\frac{P(C)}{1-P(CL(\mathbf{x}_t))} \times \frac{1}{M_{t,C}} \times \sum_{\mathbf{x}_j \in NM(\mathbf{x}_{t,C})} \|x_{t,i} - x_{j,i}\| \right) \right\} \tag{3-7}$$

En la ecuación (3-7), $CL(\mathbf{x}_t)$ retorna la etiqueta de la clase de la instancia $\mathbf{x}_t$, y $P(C)$ es la probabilidad de que las instancias pertenezcan a la clase C. $x_{t,i}$ es el valor de la característica F_i en la instancia $\mathbf{x}_t$. $NH(\mathbf{x}_t)$ denota el conjunto de muestras más cercano a $\mathbf{x}$ y con la misma clase de $\mathbf{x}$. Una muestra en $NH(\mathbf{x}_t)$ es llamada "*nearest hit*" de $\mathbf{x}$. $NM(\mathbf{x}_t, C)$ denota el conjunto de muestras más cercano a $\mathbf{x}$ y con la etiqueta de la clase C, con $(C \neq CL(\mathbf{x}_t))$. Y una muestra en $NM(\mathbf{x}_t)$ es llamada "*nearest miss*" de $\mathbf{x}$. $M_{t,CL(\mathbf{x}_t)}$ es el tamaño de $NH(\mathbf{x}_t)$, y $M_{t,C}$ es el tamaño de $NM(\mathbf{x}_t, C)$. Usualmente ambos tamaños son establecidos en una constante especificada.

El **Algoritmo 2** describe el pseudocódigo de *Relief F*, preservando la notación original de [132], el cual consiste en un bucle principal que se repite m veces, donde m corresponde al número de muestras de datos para realizar la estimación de la relevancia.

Algoritmo 2: *Relief F*

calculate prior probabilitiesP(C) for all classes;
set all weights t$W[A] := 0{,}0$;
for $i =$ **to** m **do**;
randomly select an instance R_j;
find k nearest hits H_j;
for all classes $C \neq cl(R_i)$**do**;
  from class C find k nearest misses $M_j(C)$;
end for;
for$A :=$ **1 to** a **do** H;
  $\overline{H} := -\sum_{j=1}^{k} diff\,(A, R_i, H_j)/k$
  $\overline{M} := -\sum_{C \neq cl(R_i)} \left[\left(\frac{P(C)}{1-P(cl(R_i))} \right) \sum_{j=1}^{k} diff\,(A, R_i, M_j(C)) \right] /k$;
  $W[A] := W[A] + (\overline{H} + \overline{M})/m$;
  end for;
end for;
return $\boldsymbol{W}$;

3.4.5. Prueba de Friedman

la prueba de Friedman es una prueba no paramétrica de comparación de tres o más muestras relacionadas, desarrollado por el economista Milton Friedman. Esta prueba se utiliza para

seleccionar n grupos de k elementos de forma que los elementos de cada grupo sean lo más parecidos posible entre sí, y a cada uno de los elementos del grupo se le aplica uno de entre k tratamientos, o bien cuando a cada uno de los elementos de una muestra de tamaño n se le aplican los k tratamientos. La hipótesis nula que se contrasta es que las respuestas asociadas a cada uno de los tratamientos tienen la misma distribución de probabilidad o distribuciones con la misma mediana, frente a la hipótesis alternativa de que por lo menos la distribución de una de las respuestas difiere de las demás. Para poder utilizar esta prueba las respuestas deben ser variables continuas y estar medidas por lo menos en una escala ordinal. La prueba de Friedman se puede observar en la ecuación *(*3-8).

$$F = \frac{12}{nk(k+1)} \sum_{j=1}^{k} R_j^2 - 3n(k+1) \tag{3-8}$$

Donde k es el número de observaciones o mediciones clasificadas (columnas), n es el número de sujetos (filas) y R_j es la suma de los rangos clasificados en cada columna, los números 12 y 3 son constantes. [62, 133].

3.5. Estrategias de clasificación

En el proceso de clasificación en aprendizaje de máquina, la tarea principal es tomar cada instancia o clase de un conjunto de datos y asignarlo a una clase en particular [38]. En este contexto, para este trabajo, las instancias o clases son 4 tipos de ataques llamados: *Denial of Service (DoS), Probing Attack (PA), Remote to Local (R2L) y User to Root (U2R).* La finalidad de este modelo, es utilizar un conjunto de varias técnicas de Aprendizaje de máquinas, donde se detallan las técnicas de análisis supervisado, no supervisado y semi supervisado como: SVM, KNN, y Fuzzy c-Means„ con las que se va a trabajar, donde se puedan considerar posibles clasificaciones que se realizan de manera individual y combinarlo para obtener una clasificación global que supere en rendimiento la intervención de cada método por separado para aumentar su eficiencia y mejorar la precisión, en cuanto a ataques clasificados como ataques (verdaderos positivos) y bajar la tasa de normales clasificados como ataques (falsos positivos). Para brindar una mayor robustez al sistema.

3.5.1. Máquinas de Vectores de Soporte

Las Máquinas de Vectores de Soporte (SVM, por sus siglas en inglés) es un método desarrollado por [134], para clasificación y regresión. El enfoque puede ser esquematizado de la siguiente manera [135]:

- **Separación de clases:** de forma concisa se busca un hiperplano de separación óptimo entre las dos clases al maximizar el margen entre los dos puntos más cercanos de las

clases (ver Figura **3-4**), los puntos que yacen en los límites se denominan vectores de soporte y la mitad del margen es el mejor hiperplano o de separación óptimo.

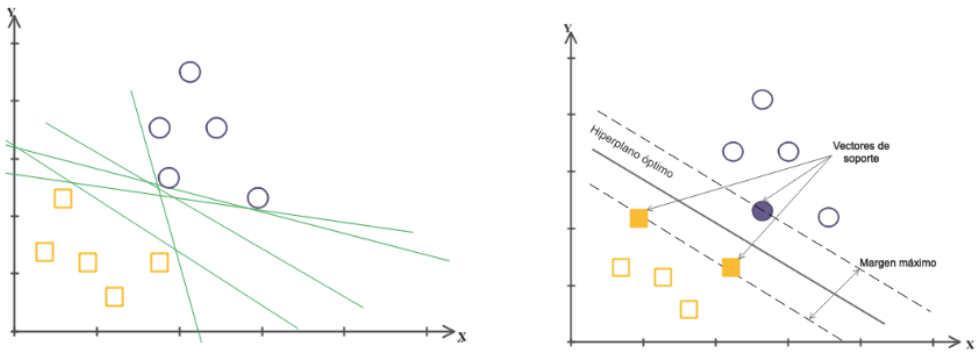


Figura 3-4.: Separación de clases SVM Fuente: [135]

- **Clases superpuestas:** los puntos de datos ubicados en el lado incorrecto del margen
- **No linealidad:** discriminante se ponderan para reducir su influencia, esto se conoce como "margen suave". cuando no se puede encontrar un separador lineal, los puntos de datos se proyectan en un espacio generalmente de mayor dimensión, donde dichos puntos se vuelven linealmente separables, esta proyección se realiza mediante la técnica de *kernel*.

Dado un conjunto T de t vectores de características de entrenamiento $x_i \in \mathbb{R}^D, i = 1, ..., t$ y las etiquetas de clase correspondientes $y_i \in \{+1, -1\}$ (para clasificación binaria). Los vectores con la etiqueta de clase $+1$ son los positivos ($clase\, C_+$), los demás pertenecen a la clase negativa ($clase\, C_-$).

3.5.2. Máquina de Vectores de Soporte Lineal

SVM lineal separa los datos en el espacio de entrada de D dimensiones con el uso de un hiperplano de decisión (ver Figura **3-5**), definido como [136]:

$$f_{(x)} : w^t x + b = 0, \tag{3-9}$$

donde w es el vector normal del hiperplano, $w \in \mathbb{R}^D$, y $\frac{b}{\|w\|}$ es la distancia perpendicular entre el hiperplano y el origen ($\|.\|$ es la 2-norma), donde $b \in \mathbb{R}$. El hiperplano es posicionado de forma tal que se maximice la distancia entre los vectores más cercanos de las clases opuestas al hiperplano.

Para dos clases linealmente separables los datos de entrenamiento deben cumplir la siguiente condición:

$$y_i \left(w^t x_i + b\right) - 1 \geq 0 \quad y_i \in \{+1, -1\} \tag{3-10}$$

Los vectores x_i para los cuales se satisfacen las igualdades de la Ecuación (3-10) se denomina vectores de soporte (ver Figura **3-4**). La distancia de estos vectores al origen se determina como $\frac{|1-b|}{\|w\|}$ y $\frac{|-1-b|}{\|w\|}$ respectivamente. Así, el objetivo de entrenar un modelo SVM es encontrar w y b para que el hiperplano separe los datos y maximice el margen $\frac{|1|}{\|w\|}$, por lo tanto, el margen teórico máximo posible de generar por el hiperplano de decisión es: $\varphi(w) = \frac{2}{\|w\|}$

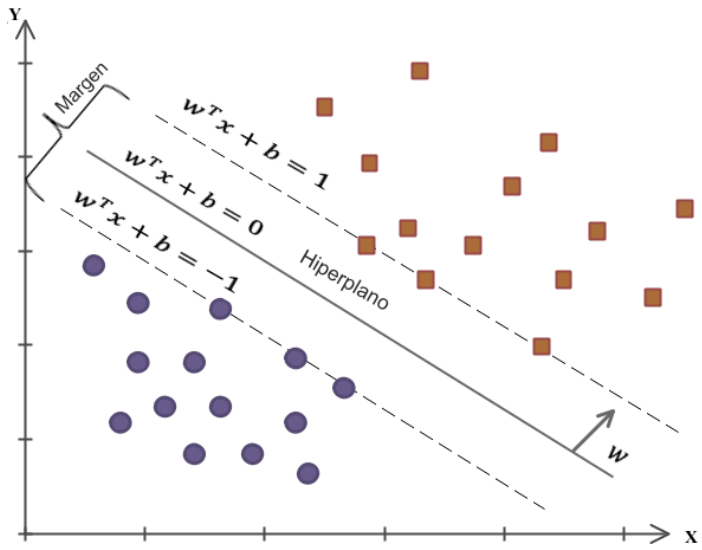


Figura 3-5.: Modelo SVM lineal (clasificación de dos clases)Fuente: [137]

Entorno a lo anterior el problema de optimización de SVM para maximizar el margen se puede definir como:

$$(w, b) = \arg\min_{w,b} \frac{1}{2} \|w\|_2^2 \quad \text{sujeto a} \quad y_i \left(w^T . x_i + b\right) \geq 1, \qquad \forall i = 1, 2, ..., N \tag{3-11}$$

Por otro lado, cuando las dos clases no son linealmente separables (por ejemplo datos ruidosos, se parecen a una clase pero son de otra), SVM determina un hiperplano con un margen muy pequeño, lo cual es muy sensible al ruido. Para resolver el problema se pueden adoptar dos técnicas, optimizar con margen suave o utilizar la técnica del *kernel* [137].

El clasificador SVM de margen suave [138] introduce una variable de holgura positiva ξi la cual es usada para registrar la cantidad de errores cometidos por el clasificador en este proceso, es decir permite que SVM cometa un cierto número de errores y mantenga el margen

lo más amplio posible para que otros puntos puedan clasificarse correctamente. Para lograr esto se minimiza la norma Euclidiana de w, lo cual corresponde a los coeficientes que definen el hiperplano, generando el problema de minimización:

$$\begin{gathered}\text{mín}_{w,b,\xi}\ \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \quad \text{sujeto a} \quad y_i\left(w^T\cdot x_i + b\right) \geq 1,\xi \quad i = 1,...,N, \\ \xi_i \geq 0, \qquad\qquad i = 1,...,N\end{gathered} \tag{3-12}$$

donde C es el parámetro de regularización para controlar la compensación entre maximizar el margen y minimizar el error de entrenamiento o penalización de holgura (cuanto mayor sea el valor de C, mayor será la penalización de los errores). Así, la formulación del problema de optimización de SVM de margen suave se determina como:

$$\begin{gathered}(w,b,\xi) = \arg\text{mín}_{w,b,\xi}\ \tfrac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N}\xi_i \quad \text{sujeto a} \quad 1-\xi_i - y_i\left(w^T\cdot x_i + b\right) \geq 1, \\ \forall i = 1,2,...,N, \xi_i \geq 0, C > 0\end{gathered} \tag{3-13}$$

3.5.3. Máquina de Vectores de Soporte no lineal

El método *kernel* es la extensión de la formulación de la Ecuación (3-13) a problemas que no tienen solución lineal y requieren una función de decisión no lineal, a fin de obtener un hiperplano SVM no lineal [139, 140]. Consiste en definir una función *kernel* que calcula el producto interno de dos vectores de características en un espacio de características no lineal derivado [141]:

$$K\left(a,a'\right) = \phi(a)^T\phi(a'),$$

donde $\phi : \mathbb{R}^D \to \mathbb{F}$ es un mapeo o transformación de un vector a de la entrada con grado d a un espacio característico no lineal (posiblemente infinitamente dimensional), donde la expansión contiene términos sólo positivos:

$$K(p,q) = (a + p^Tq)^d = a^d + da^{d-1}(p^Tq) + ... + (p^Tq)^d \tag{3-14}$$

Se puede demostrar que un *kernel* de Función de Base Radial, (RBF, por sus siglas en inglés) o *kernel* Gaussiano, tiene una expansión infinita de términos no negativos, de manera que proyecta una entrada a un espacio de características no lineal $\mathbb{F}$, de mayor dimensionalidad, donde los vectores son linealmente separables, y $K : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$. El *kernel* no requiere el cálculo del mapeo ϕ explícitamente. La función SVM no lineal se expresa como:

$$f(a) = sgn\left(\sum_{i=1}^{t}\alpha_i y_i K(x_i^T a) + b\right) \tag{3-15}$$

donde α_i es un multiplicador Lagrange. En la (Figura **3-6**), se ilustra un ejemplo del mapeo de un conjunto de datos de un espacio bidimensional a uno de dimensión superior. En el

espacio de entrada original, los vectores de características que pertenecen a las dos clases no son linealmente separables, pero cuando el espacio es transformado permite determinar un hiperplano que separa los vectores. Considerando la función *Kernel* $K(p,q) = \phi(p)^T\phi(q)$

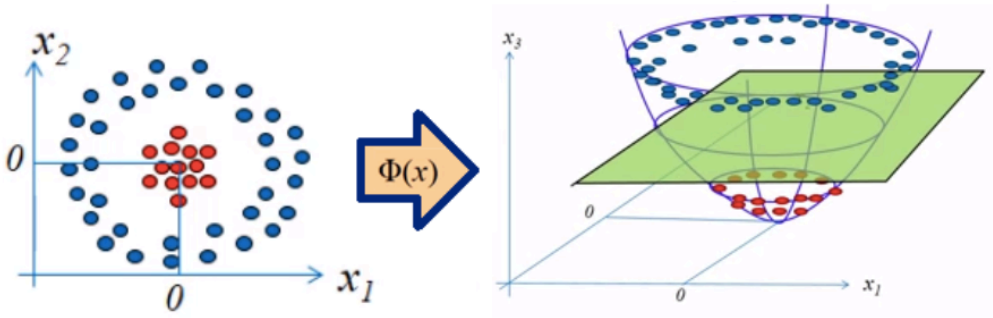


Figura 3-6.: Mapeo de conjunto de datos de un espacio bidimensional a un espacio tridimensional. Fuente:[142]

donde p y q son vectores, los *kernel* comúnmente empleados con SVM incluyen [135] [142]:

- *Kernel* lineal definido como, $K(p,q) = p^Tq$,
- *Kernel* polinomial con grado d, donde la expansión contiene términos sólo positivos: $K(p,q) = (a + p^Tq)^d = a^d + da^{d-1}(p^Tq) + ... + (p^Tq)^d$
- *Kernel* de Función de Base Radial, (RBF, por sus siglas en inglés) o *Kernel* Gaussiano, en la cual también se puede demostrar que tiene una expansión infinita de términos no negativos: $K(p,q) = exp(-\beta \left\|p - q\right\|^2)$

3.5.4. Máquina de Vectores de Soporte multiclases

Los enfoques clásicos construyen el clasificador multiclase como la combinación de N tareas de clasificación binaria independientes (es decir resuelven por separado las tareas correspondientes) [143]. La traducción de multiclases a clases binarias es desarrollada a través de diferentes esquemas, donde los más comunes son uno contra uno y uno contra todos. Las funciones de decisión de SVM resultantes son consideradas en su totalidad y la clase para cada muestra en el conjunto de prueba se decide mediante el esquema correspondiente. Sea $S_c = \{(x_1, y_1), (x_2, y_2), ..., (x_i, y_i)\}$ un conjunto de entrenamiento donde, $x_i \in R^m$ y $y_i \in \{1, 2, ..., k\}$ [136, 143–145]:

- **Esquema uno contra todos:** Esta técnica construye k SVM clasificadores, de modo que para un problema de $k-$clases se requiere determinar los k hiperplanos. Cada i^{th} clasificador SVM considera todas las muestras de entrenamiento etiquetadas con i como positivas y todas las restantes como negativas, (ver Figura **3-7**) [146]. El objetivo de cada i^{th} clasificador SVM es determinar los coeficientes w y b óptimos del hiperplano

de decisión, de modo que se puedan separar las muestras con el resultado i de todas las otras muestras en el conjunto de entrenamiento, tal que:

$$\begin{cases} \text{encontrar } w^i \text{ y } b^i \text{ para minimizar } \frac{\|w^i\|^2}{2} + C\sum_{j=1}^{m}\xi_j^i \\ \text{sujeto a } \; y_j(w^i \cdot x_j - b^i) \geq 1 - \xi_j^i, \xi_j^i \geq 0, \text{ para todo } j = 1, 2, ...m. \end{cases} \tag{3-16}$$

Luego de determinar todos los hiperplanos con SVM binaria, la clase para una muestra de prueba x viene dada por la categoría que tiene el valor máximo para la función de aprendizaje, así: $clase\,(x') = \arg\max_i 1, 2, ..., k(w^i \cdot \phi(x')) - b^i)$

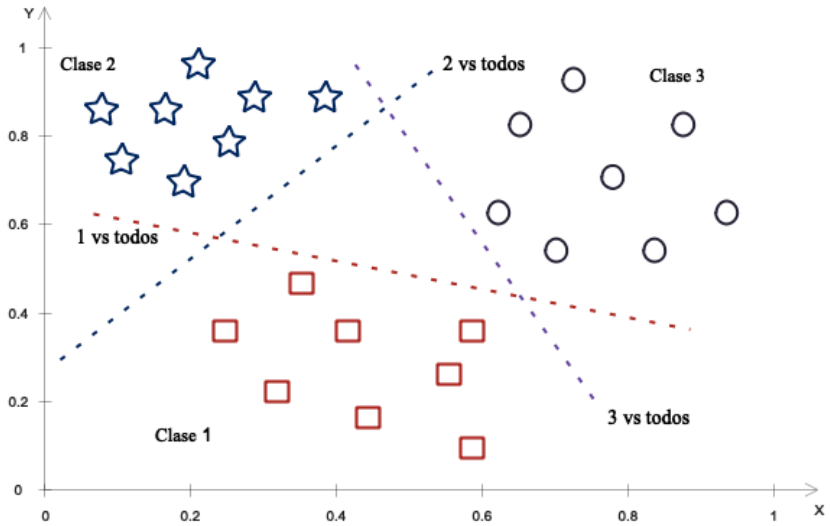


Figura 3-7.: SVM multiclases (uno *vs* todos) Fuente:[146]

- **Esquema uno contra uno:** Esta técnica construye $\frac{k(k-1)}{2}$ SVM clasificadores para un problema de k clases. Cada i^{th} clasificador SVM es entrenado sobre datos de cada dos clases, i y j, donde las muestras etiquetadas con i son consideradas positivas mientras que las de la clase j se toman como negativas, ver Figura **3-8** [146]. El objetivo de cada i^{th} clasificador SVM es determinar los coeficientes óptimos del hiperplano de decisión para discriminar las muestras con resultado i de las muestras con resultado j, así:

$$\begin{cases} \text{encontrar }, w^{ij} \; y \; b^{ij} \text{ para minimizar } \frac{\|w^{ij}\|^2}{2} + C\sum_{l=1}^{m}\xi_j^{ij} \\ \text{sujeto a } \; y_l(w^{ij} \cdot x_l - b^{ij}) \geq 1 - \xi_l^{ij}, \xi_l^{ij} \geq 0, \text{ para todo } l = 1, 2, ...m. \end{cases} \tag{3-17}$$

Cuando los hiperplanos $\frac{k(k-1)}{2}$ son encontrados, para determinar la clase para una muestra de prueba x se utiliza el mayor voto a distancia.

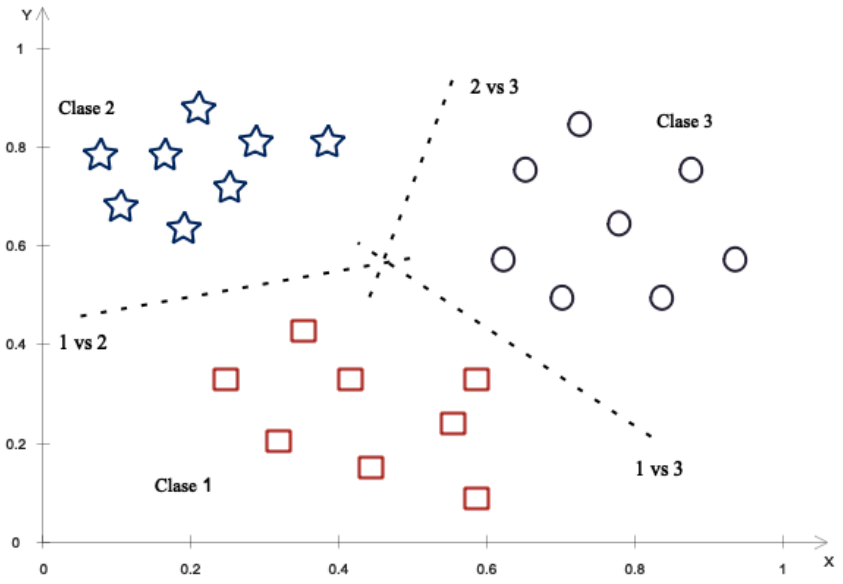


Figura 3-8.: SVM multiclases (uno *vs* uno)Fuente:[146]

3.5.5. *Fuzzy c-Means* Semi-Supervisado

El *Fuzzy c-Means* (FCM), propuesto por Dunn [147] y Bezdek [148], es un método de *clustering* que asigna a cada dato un valor de pertenencia o función de membresía dentro de cada *cluster*, lo cual le permite a un dato específico pertenecer parcialmente a dos o más agrupaciones. Además FCM es un algoritmo de aprendizaje no supervisado, es decir realiza la clasificación sin utilizar la información de la etiqueta de clase [149, 150].

El algoritmo *Fuzzy c-Means* Semi-Supervisado (SSFCM, por sus siglas en inglés) utiliza dos componentes supervisada y no supervisada para guiar el agrupamiento a una solución significativa [151]. Así, el algoritmo usa los datos etiquetados de la entrada como ejemplos de entrenamiento para clasificar datos no etiquetados, lo cual implica el cálculo iterativo de los centros de clúster y la matriz de partición para minimizar la función objetivo hasta que se cumpla un criterio de terminación [152].

La forma de clasificar los n números de datos se plantea de la siguiente manera [151,153,154]:

$$X = \left\{x_k | x_k = (x_{k1}, ..., x_{kp})^T \in \Re^p, k = 1 \sim n\right\}$$

En $\mathbf{c}$ números de *clusters*:

$$\mathbf{C} = \{\mathbf{C}_i | i = 1 \sim \mathbf{c}\}$$

Los cuales son representados como centroides:

$$\mathbf{V} = \left\{ v_i \,|\, v_i = (v_{i1}, ..., v_{ip})^T \in \Re^p, i = 1 \sim \mathbf{c} \right\} \tag{3-18}$$

En un espacio patrón $\Re^p$ y la norma Euclidiana se define en el espacio como:

$$\|x_k - v_i\| = \sqrt{\sum_{j=1}^{p} (x_{kj} - v_{ij})^2} \tag{3-19}$$

$u_{ki} \in [0, 1]$ representa el grado de membresía que x_k pertenece a $\mathbf{C}_i$ y el objetivo de esta clasificación es obtener la siguiente $\mathbf{U}$:

$$\mathbf{U} = \left\{ u_{ki} \,|\, u_{ki} \in [0, 1], \sum_{i=1}^{c} u_{ki} = 1, \quad k = 1 \sim n, \, i = 1 \sim \mathbf{c} \right\} \tag{3-20}$$

$\overline{u}_{ki} \in [0, 1]$ representa el grado de membresía supervisado que x_k pertenece a $\mathbf{C}_i$ y

$$\overline{\mathbf{U}} = \{ \overline{u}_{ki} \,|\, \overline{u}_{ki} \in [0, 1], \, k = 1 \sim n, \, i = 1 \sim \mathbf{c} \} \tag{3-21}$$

Se introduce el grado de membresía supervisado $\overline{u}_{ki}$ dentro de la función FCM, minimizando su función objetivo:

$$\mathbf{J}(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^{n} \sum_{i=1}^{c} |u_{ki} - \overline{u}_{ki}|^m \left\| x_k - v_i \right\|^2, \tag{3-22}$$

Se obtiene la siguiente solución óptima:

$$\mathbf{v_i} = \frac{\sum_{i=1}^{n} |u_{ki} - \overline{u}_{ki}|^m x_k}{\sum_{i=1}^{n} |u_{ki} - \overline{u}_{ki}|^m} \tag{3-23}$$

Y la función de membresía se determina como:

$$\begin{cases} \overline{u}_{ki} + 1 \sum_{i=1}^{c} \overline{u}_{ki}, & (i = arg\, min_l\, d_{kl}) \\ \overline{u}_{ki}. & (de\, otra\, manera) \end{cases} \tag{3-24}$$

A continuación se presenta el **Algoritmo 3** con base en las soluciones óptimas descritas anteriormente, este obtiene u_{ki} y v_i minimizando J por optimización iterativa, así como FCM.

3.5.6. k-Vecinos más Cercanos

Los vecinos más cercanos a (k-NN, por sus siglas en inglés) es un método de agrupación en clúster bien conocido. Se basa en encontrar similitudes en los puntos de datos, o por lo que se puede llamar similitud de características [155], que puede utilizarse para resolver problemas de clasificación y regresión. El algoritmo k-NN clasifica una muestra de prueba

Algoritmo 3: Algoritmo Fuzzy C-Means Semi-Supervisado

Inicializa c, matriz de membresia etiquetada X y matriz de membresia inicial U^0
Calcula centros de cluster usando la ecuación (3-21)
Calcula matrices de covarianza difusa
Calcula distancias cuadradas entre los centros de agrupación y los patrones de datos
Actualiza la matriz de particiones **U** usando la ecuación (3-23)
if $\left\|\overline{\mathbf{U}} - \mathbf{U}\right\| < \epsilon$, para. De lo contrario vaya a la linea 2 con $\mathbf{U} = \overline{\mathbf{U}}$

sin etiquetar con base en las medidas de similaridad entre los vecinos o puntos más cercanos k. La distancia entre la muestra de prueba y cada una de las muestras de entrenamiento están determinadas por una medida de distancia específica [156].

Dada una nueva observación x_0, el algoritmo k-NN determina su etiqueta de clase de la siguiente manera [76]:

- Dentro de la muestra de entrenamiento, se selecciona k puntos de datos ($k = 1, 2, ...$), tal que son los más cercanos de x_0, los cuales se conocen como los vecinos más cercanos a k. La cercanía se evalúa por medio de alguna medida de distancia, por ejemplo, distancia Euclidiana, distancia Chebychev o distancia Manhattan.

- La etiqueta de clase de x_0 está determinada por un voto mayoritario de sus vecinos más cercanos a k, es decir, x_0 se asigna a la clase que es más frecuente entre sus vecinos más cercanos a k.

El **Algoritmo 4** describe los pasos principales de este método [157]:

Algoritmo 4: Algoritmo k Vecinos mas Cercanos k-NN

Definir k **while** (no se cumple el criterio de parada) **do**
Calcular distancias desde otros puntos de datos hasta el punto i
Las distancias calculadas
Los k puntos con las distancias más pequeñas
Asignar el punto de prueba a la clase por mayoría simple
Devolver la clase
end while

La (Figura **3-9**) ejemplifica la tarea de clasificación de k-NN, donde hay dos clases, estrella y diamante . La tarea consiste en determinar a qué clase pertenece triángulo . Si se usa $k = 3$

(el círculo discontinuo más pequeño), entonces triángulo debe pertenecer a la clase diamante, pero si se usa $k = 7$ (el círculo sólido más grande) entonces triángulo debe ser clasificado como estrella. Sin embargo si $k = 6$ (el círculo punteado) sería difícil clasificar a triángulo porque esto conduce a un empate entre las dos clases estrella y diamante, lo cual resalta la importancia de elegir el parámetro correcto para k y su sensibilidad. Por tanto valores muy pequeños de k dan lugar a ajustes excesivos y alta sensibilidad al ruido, mientras valores grandes de k pueden llevar a un mayor sesgo y menor precisión dada la inclusión de muestra que no son vecinos cercanos reales.

Por ello una guía básica para elegir k es $k < \sqrt{n}$ para un conjunto dado de n puntos de datos de entrenamiento [157].

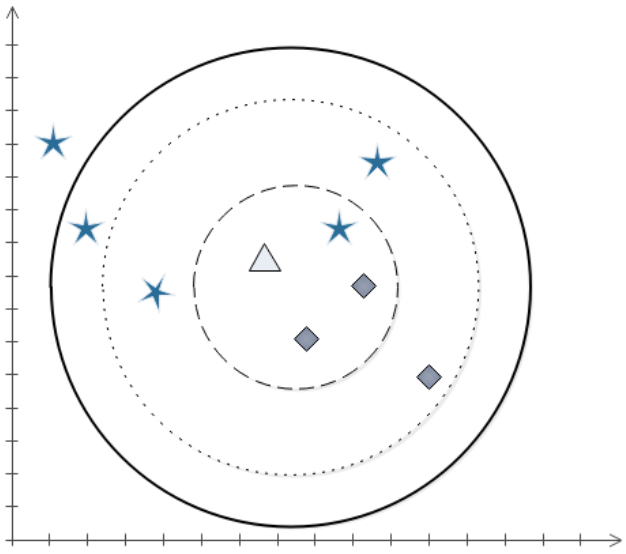


Figura 3-9.: Representación esquemática de k-NN. Fuente:[158]

4. Marco Experimental

En este capitulo se presenta la metodología propuesta en la Figura **4-1**, donde este proceso es offline y todas las mediciones de los ataques ya lo tiene la base datos.

donde se describe la base de datos y se expone el modelo propuesto para la generación de los espacios de representación, la representación efectiva, donde se encuentra la selección de características para reducir la dimensionalidad de los diferentes tipos de ataques, además de la estructuración metodológica que permita la detección de los cuatro tipos de ataques.

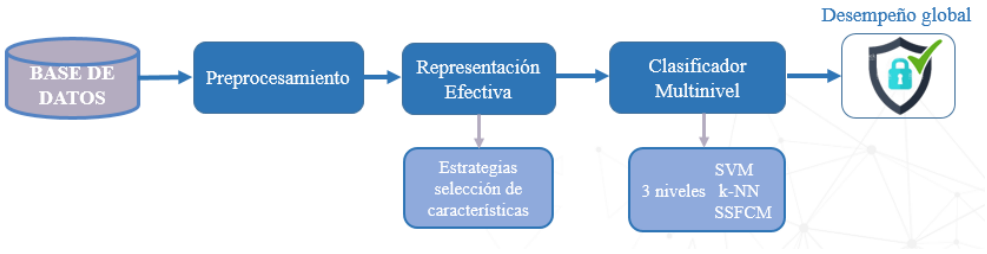


Figura 4-1.: Metodología

4.1. Base de datos: KDD-Cup99

El conjunto de datos KDD-Cup99 es ampliamente reconocido como el estándar o punto de referencia para la evaluación de IDS en temas de minería de datos y selección de características [158]. Es una modificación del conjunto de datos DARPA98 bajo el patrocinio de la Agencia de Proyectos de Investigación Avanzada de Defensa (DARPA), el Laboratorio de Investigación de la Fuerza Aérea (AFRL), y el MIT Lincoln Labs [159]. La base de datos consta de 4,898,430 registros, 41 características que describen el tipo de conexión y un vector clases que para 4 clases distingue 22 formas de ataques diferentes y una clase con etiqueta normal.

Las 5 clases que contiene la base de datos son [160]:

- Denegación de Servicio (DoS, por sus siglas en inglés): El delincuente informático intenta evitar que los usuarios legítimos ingresen o usen un servicio. El resultado de este

ataque es la falta de disponibilidad de recursos, es decir, los recursos están demasiado ocupados o demasiado llenos para atender solicitudes de red legítimas.

- *Probing Attack* (PA): El atacante intenta obtener información sobre el host de destino con el propósito aparente de burlar sus controles de seguridad.
- Ataque Remoto a Local (R2L, por sus siglas en inglés): Ocurre cuando el delincuente tiene la capacidad de enviar paquetes a una máquina de forma remota, sin tener una cuenta de usuario en esa máquina y explora alguna vulnerabilidad para tener acceso como un usuario local.
- Ataque a usuario raíz (U2R, por sus siglas en inglés): El atacante explora las vulnerabilidades del sistema con una cuenta de usuario normal (posiblemente obtenida mediante el rastreo de contraseñas, un ataque de diccionario o ingeniería social), para adquirir privilegios de administrador (acceso raíz al sistema).
- La clase Normal: para este trabajo, se entiende como el perfil de comportamiento del tráfico que adopta un modelo dentro de los rangos de tráfico legítimo. La construcción del perfil puede ser estática o dinámica. Para el modo estático, se construye un perfil que solamente se reemplaza cuando se cambia la red, mientras en el modo dinámico, el perfil se actualiza de acuerdo con los cambios de comportamiento de la red.

Las formas de ataque correspondientes a las cuatro clases mencionadas anteriormente, se muestran en la Tabla **4-1**, donde se pueden evidenciar con más detalle cada forma de ataque en [161,164,165].
En la Figura **4-2** se puede observar el desbalance entre las clases, así como las formas de ataque, representando el mayor porcentaje de la clase DoS: "smurf" y "neptune", mientras la clase Normal representa el 20 % y las 20 formas restantes sólo constituyen el 1 % de los datos.
De otro modo, las características que describen el tipo de conexión se agrupan en cuatro categorías que son: Básicas (B), Contenido (C), Trafico (T) y Host (H) [161].

- **Básicas:** Encapsula todos los atributos que se pueden extraer de una conexión TCP/IP. Se obtienen del encabezado del paquete, sin examinar el contenido del paquete (duración, tipo de protocolo, servicio, marca y el número de bytes enviados desde el origen al destino y viceversa).
- **Características de contenido:** Se obtienen analizando el contenido del paquete TCP, lo cual incluye características tales como el número de intentos de inicio de sesión fallidos.
- **Características de tráfico:** Son los atributos calculados utilizando una ventana de tiempo de dos segundos. Determinan la duración de la conexión desde una dirección

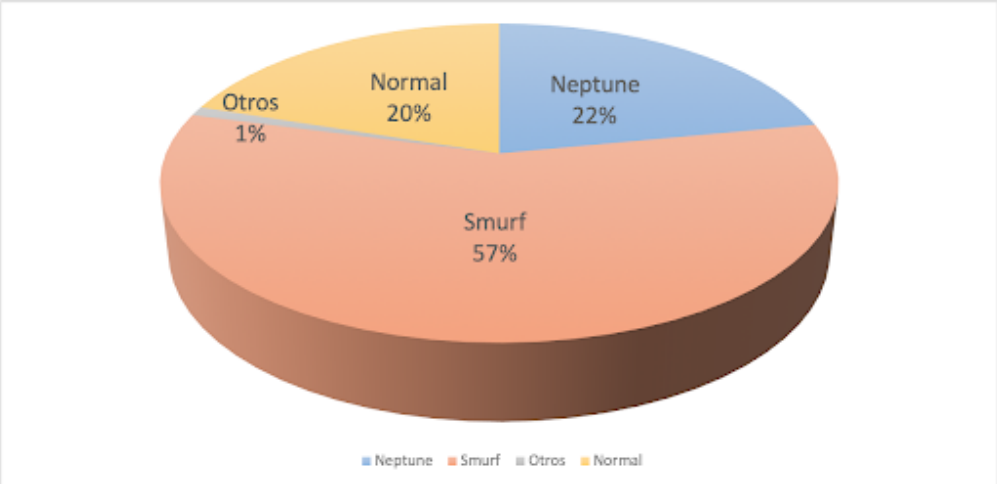


Figura 4-2.: Distribución de 23 formas en tipos de ataques y normal.

IP de origen a las direcciones IP de destino. La conexión es una secuencia de paquetes de datos que comienzan y terminan en algunos momentos predefinidos.

- **Características de Host:** Son los atributos diseñados para evaluar ataques que duran más de dos segundos. Utilizan una ventana histórica estimada sobre el número de 100 conexiones (no intervalos de tiempo), lo cual es adecuado para describir ataques que duran más que el intervalo de las características de tiempo estipuladas [158, 160–162].

La Tabla **4-2**, muestra 41 características que conforman la base de datos KDD Cup 99 de acuerdo con su categoría, donde se puede ver con más detallé en: [160, 163, 164].

4.2. Preprocesamiento y representación efectiva

Para obtener una matriz de características apta para el clasificador multinivel, inicialmente se realiza un preprocesamiento y luego se genera un nuevo espacio de representación de características aplicando un esquema de selección de características cuando llego hacer requerido.

4.2.1. Preprocesamiento

El preprocesamiento de los datos consistió en depurar la información útil, de datos anómalos que no aportaban información y ajustar la representación de los datos, bajo las siguientes ejecuciones derivadas de técnicas de análisis estadístico multivariado:

1. Remoción de muestras con valores inválidos, eliminando toda la dimension, como datos censurados, valores alfabéticos.

2. Mapeo de las columnas expresadas en tipo carácter a un valor único por cada string, los cuales contenían la información respectiva al tipo de protocolo, servicio, bandera y forma de ataque.

3. Normalización de cada característica de la base de datos en el rango [0,1], utilizando escalado de características dentro del rango mínimo-máximo, debido a la gran variación entre los valores, como se describe en la ecuación 4-1:

$$\chi' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{4-1}$$

donde X_{min} y X_{max} hacen referencia al valor mínimo y máximo de cada característica respectivamente. Las características nominales protocol type, Service and flag, se le asignó un número a cada uno de los string, y este valor fue normalizado de acuerdo a lo descrito anteriormente.

4. Se eliminaron aquellas formas de ataque con menos de 9 muestras, ftp_write, phf, multihop y spy de la clase R2L, perl de la clase U2R, obteniendo una base de datos final con cuatro ataques, 17 formas de ataque y la clase normal. Este proceso se llevó a cabo dado que estas formas de ataque no aportan información suficiente para construir el clasificador.

4.2.2. Representación efectiva

Se realizaron pruebas de capacidad descriminante en la base de datos para verificar la utilización de usar técnicas de representación efectiva, la generación de un nuevo espacio de representación fue implementada mediante un esquema de selección de características utilizando los métodos FRS y *Relief F* descritos en la sección 3.4, con el objetivo de determinar las características que aportan información al sistema, tienen mayor capacidad discriminante y descartar las características redundantes y/o irrelevantes, lo cual permite mejorar la precisión del clasificador multinivel, así como reducir el costo computacional al disminuir la dimensionalidad de la base de datos.

Inicialmente se plantea la selección de características relevantes utilizando el método FRS, dadas sus bondades para trabajar con datos mixtos (reales - discretos) y ruidosos, lo cual es confluente con la base de datos KDD-Cup99 que presenta una alta redundancia del 78 %, existiendo sólo 1,074,992 puntos de datos únicos [162]. Al ser una base de datos considerablemente desbalanceada y de gran dimensión, su alto costo computacional sólo permitió trabajar con 1000 muestras aleatorias de cada forma de ataque, sin embargo se trabajó con la totalidad de los datos para aquellas formas de ataque con menos de 1000 muestras. De este modo, la selección de características basada en FRS se realizó con 41 características y 10.420 muestras, representando únicamente el 0.21 % de la base de datos. Se realizó la técnica de

muestreo de forma aleatoria, con el fin de garantizar que los datos no se sesgarán a datos específicos.

El método *Relief F* se llevó a cabo en dos niveles, usando $k = 10$ de acuerdo con lo reportado en la literatura. Inicialmente se carga toda la base de datos y se selecciona las muestras de las clase R2L, U2R y Normal, para un total de 41 características y 973.934 muestras lo que representa el 20 % de la base de datos. Posterior a ello se implementa el mismo método para las formas de R2L y U2R en el nivel tres, para un total de 41 características y 1.152 muestras. Para las clases DoS y Probing Attack no se implementó el selector*Relief F* debido al buen desempeño de los clasificadores sin selección

4.3. Clasificador multinivel

El clasificador multinivel se llevó a cabo en tres niveles para clasificar diferentes formas de ataque de DoS, Probing Attack, R2L y U2R (ver Figura **4-3**), con el objetivo de obtener un mejor desempeño.

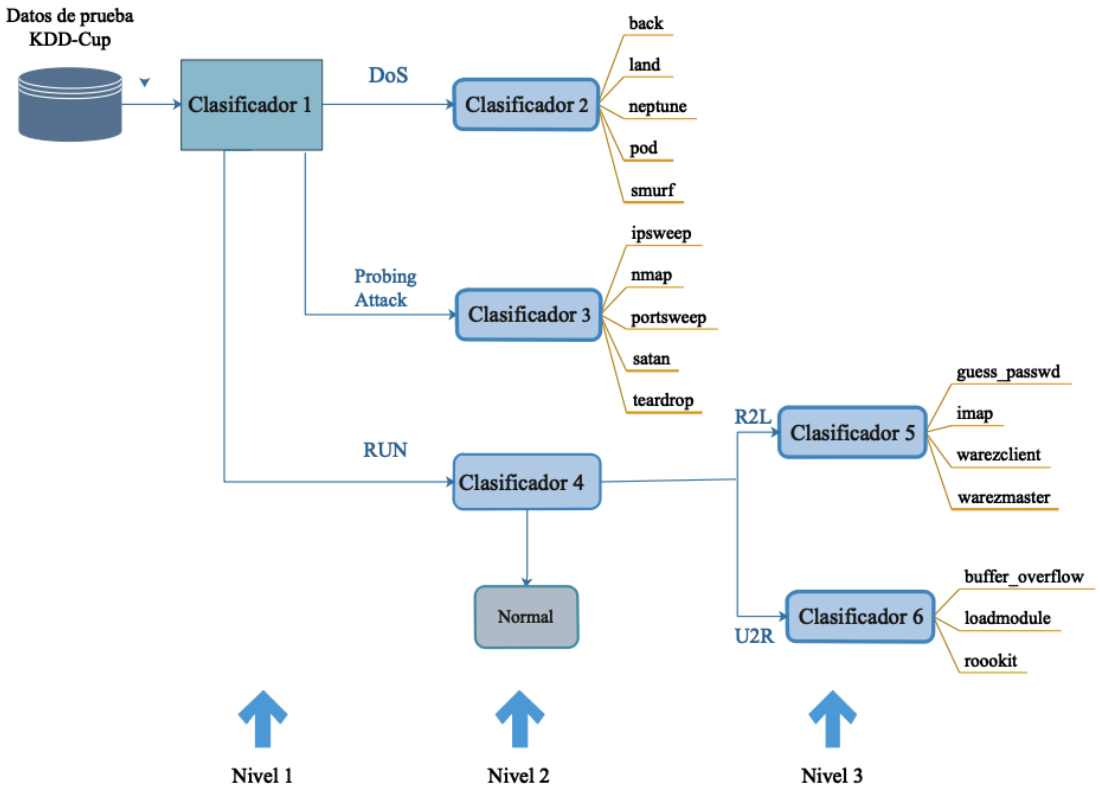


Figura 4-3.: Clasificador Híbrido Multinivel

Posteriormente, se procedió a realizar pruebas de forma independiente de cada clasificador, es decir que se entrena y se evalúa el desempeño únicamente teniendo en cuenta los datos del nivel. Se usaron los métodos k-NN, SSFCM y SVM, descritos en la sección 3.5. SVM funciona muy bien correlacionando datos en un espacio de características de grandes dimensiones, donde se busca los vectores en los cuales debe soportar los hiperplanos. Así mismo se llevó a cabo con el método de k-NN, que se basa en clasificar cada dato nuevo en la clase que corresponda, de acuerdo a los K vecinos mas cerca entre una clase y la otra. Por último se implementó el método SFCM, básicamente esta técnica de agrupación de datos se centra en que cada punto de datos pertenece a un clúster con un grado de pertenencia para cada uno , el objetivo de este método está en agrupar objetos similares y separar los diferentes. El desempeño fue evaluado en términos de la precisión, con el fin de determinar que algoritmo de clasificación se comporta mejor en cada uno de los niveles.

Nivel 1 - clasificador DoS, Probing Attack y RUN: Como se describe en la (Figura **4-4** (a)), el nivel 1 consta de un clasificador que reconoce 3 clases, DoS, Probing Attack y una combinación de las clases R2L, U2R y Normal, la cual se denomina para esta tesis como RUN.

Nivel 2 - Clasificador formas de DoS, formas de Probing Attack, ataques R2L, U2R y Normal: El nivel 2 consta de tres clasificadores, el primero reconoce las diferentes formas de DoS, el segundo las diferentes formas de Probing Attack y el tercero diferencia en los ataques R2L, U2R y Normal (ver Figura **4-4** (b)).

Nivel 3 - Clasificador formas de R2L, formas de U2R: Nivel 3 lo integran dos clasificadores que reconocen las diferentes formas de R2L y U2R (ver Figura **4-4** (c)).

Error global del Clasificador Multinivel: Para el cálculo del error global en el Clasificador Multinivel y el error en los tres niveles, se tomó la ecuación **4-2.**

$$E\,G_i = W_i \cdot E\,R \tag{4-2}$$

Donde $W_i = \frac{N_{ci}}{N_{ct}}$ y $ER_i = \frac{FP_i}{FP_i + VP_i}$

W_i: Es la ponderación de de la clase i

N_{ci}: Es el número de elementos de la clase i

N_{ct}: Es el número total de elementos para todo el nivel determinado

ER_i: Es el error Relativo de la clase i

FP_i: Son los falsos positivos asociados a la clase i

VP_i: Son los Verdaderos positivos asociados a la clase i

EG_i: Es el Error Global de la clase i

Respecto a los parámetros de los métodos de clasificación empleados se tiene que, el algoritmo de SVM fue utilizado en conjunto con un kernel de base radial de la forma $e^{-\gamma|u-v|^2}$, el parámetro γ del Kernel, de manera que después de la ejecución de un conjunto de pruebas de minimización del riesgo, se logró la sintonización del parámetro del Kernel en $0,8$ y el parámetro de costo del SVM en $4,5$. Por su parte el k-NN empleó los 5 vecinos más cercanos para realizar la clasificación. Finalmente, el parámetro de Fuzificación m para SSFCM (ver ecuación **3-23**) fue de 1,1.

Finalmente, una vez determinado el mejor algoritmo en cada nivel, se procedió a realizar pruebas utilizando de forma conjunta el clasificador multinivel, las pruebas fueron llevadas a cabo, utilizando todas las dimensiones, las características seleccionadas por FRS y las seleccionadas por *Relief F*.

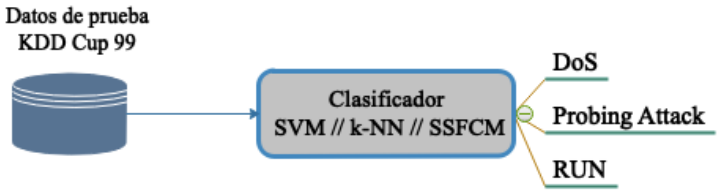
Tabla 4-1.: Clases y formas de ataque de la base de datos KDD CUP 99

Tipo	**Forma**	**Número de muestras**
DoS	back	2203
	land	21
	neptune	1072017
	pod	264
	smurf	2807886
Probing Attack	ipsweep	12481
	nmap	2316
	portsweep	10413
	satan	15892
	teardrop	979
R2L	guess_passwd	53
	pftp_write	8
	imap	12
	phf	4
	multihop	7
	spy	2
	warezclient	1020
	warezmaster	20
U2R	buffer_overflow	30
	pearl	3
	loadmodule	9
	rootkit	10

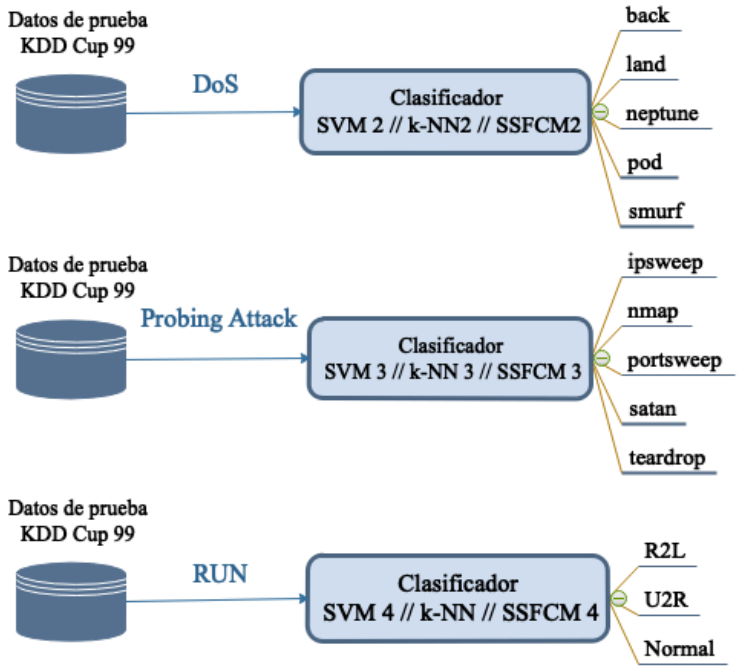
Tabla 4-2.: Características de la Base de Datos

No.	Características	Categorías
1	Duration	B
2	protocol_type	B
3	Service	B
4	Flag	B
5	Src_bytes	B
6	Dst_bytes	B
7	land	B
8	Wrong_fragment	B
9	Urgent	B
10	Hot	C
11	Num_failed_logins	C
12	Logged_in	C
13	Num_compromised	C
14	Root_shell	C
15	Su_attempted	C
16	Num_root	C
17	Num_file_creations	C
18	Num_shells	C
19	Num_access_files	C
20	Num_outbound_cmds	C
21	Is_host_login	C
22	Is_guest_login	C
23	Count	T
24	Srv_count	T
25	Serror_rate	T
26	Srv_serror_rate	T
27	Rerror_rate	T
28	Srv_rerror_rate	T
29	Same_srv_rate	T
30	Diff_srv_rate	T
31	Srv_diff_host_rate	T
32	Dst_host_count	H
33	Dst_host_svr_count	H
34	Dst_host_same_svr_rate	H
35	Dst_host_diff_svr_rate	H
36	Dst_host_same_src_port_rate	H
37	Dst_host_same_src_port_rate	H
38	Dst_host_serror_rate	H
39	Dst_host_svr_serror_rate	H
40	Dst_host_serror_rate	H
41	Dst_host_svr_serror_rate	H

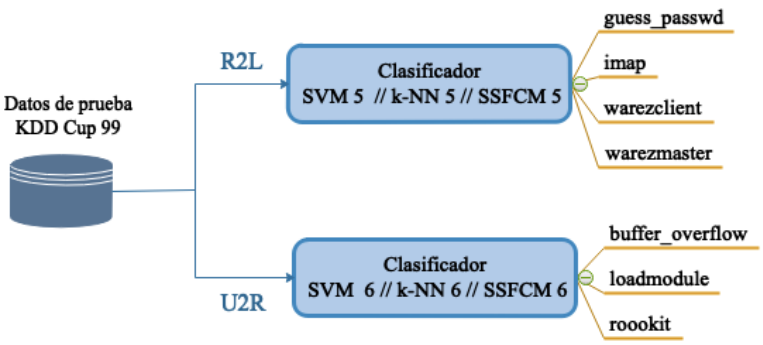
[1] Básicas (B), Contenido (C), Trafico (T) y Host (H)



(a) Nivel 1: clasificador, DoS, Probing Attack y RUN



(b) Nivel 2: Clasificador formas de DoS, formas de Probing Attack, ataques R2L, U2R y Normal



(c) Nivel 3: Clasificador formas de R2L, formas de U2R

Figura 4-4.: Descripción clasificador multinivel

5. Resultados y discusión

El software de desarrollo que se usó en esta tesis es Matlab R2018a, con el cual se desarrollaron los programas para la implementación de los modelos y la realización de las pruebas, obteniendo como tiempo para cada entrenamiento alrededor de 68 horas, en un equipo con una arquitectura que se evidencia en la tabla**5-1**. Es importante mencionar que el tiempo de entrenamiento es muy alto, pero después que los algoritmos están entrenados, el tiempo de ejecución para la verificación de si una muestra es ataque o no, es de 50 milisegundos.

Tabla 5-1.: Especificaciones del equipo

LENONVO THINK-CENTRE	
Procesador	Intel® Core(TM) i7-7700 CPU 3.60 GKz
Memoria RAM	32 GB
Disco duro SSD	256 GB
Disco duro HDD	1 TB
Sistema Operativo	Windows 10 Pro
Software	Matlab R2018a

Para dar cumplimiento al objetivo uno de la tesis, se tomó la base de datos KDD-Cup99 para las pruebas de entrenamiento y validación de los algoritmos.

a. Debido a que en la literatura era muy frecuente encontrar que los trabajos sobre la base de datos KDD-Cup99 reportaban la necesidad de generar un nuevo espacio de representación derivada de la misma base de datos, y en los reportes publicados se observó que efectivamente las pruebas eran realizadas sobre una base de datos modificada, se presumió la necesidad de hacerlo también, y esa es la razón por la que se incluyó en el primer objetivo específico de la tesis, suponiendo realizarlo desde la misma base de datos usando técnicas de extracción de características, como por ejemplo las representaciones de base ortogonal.

b. En esa medida, al momento de iniciar la ejecución de la tesis, se realizaron como primeras pruebas, el análisis del poder discriminante de los atributos estimados y contenidos en la base de datos KDD-Cup99. Donde, en caso de obtener resultados bajos de discriminación, se debía proceder al uso de técnicas de representación ortogonal para encontrar la com-

binación lineal de variables no correlacionadas que permitieran diferenciar o discriminar mejor las clases.

c. Para este fin, se realizaron diferentes pruebas de clasificación en los tres niveles, donde se encontró que el problema más grande radicó en el tamaño de la base de datos, dado que contiene más de 4 millones y medio de datos por 41-dimensiones.

d. Inicialmente, se hizo el preprocesamiento de los datos, que consistió en una limpieza de información, explicado paso a paso en el documento de la tesis.

e. Luego, se utilizaron los métodos de selección FRS y Relief-F, para verificar a ciencia cierta el poder discriminante de la base de datos, tomando en consideración las diferentes clases existentes.

f. Se encontró que, para el primero y segundo nivel de clasificación, la selección de características bajaba el porcentaje de acierto de los clasificadores. Por lo que se determinó que la totalidad de las características aportaba a las necesidades de clasificación para el nivel uno y dos, con lo que se evidenció que la generación de un nuevo espacio de representación no era completamente necesario para estos dos niveles, y sería injustificado el costo computacional teniendo en cuenta el gran tamaño inicial del espacio disponible por la base de datos.

g. Adicionalmente, para el nivel tres, se encontró que Relief-F fue el método que arrojó mejor desempeño, ayudando a superar la baja variabilidad entre clases de este nivel y aportando al poder discriminante.

h. Así, entonces, se procedió a la generación de un espacio de representación reducido o modificado con menor dimensión, para la tarea de clasificación del nivel tres.

i. De esta manera, se evidenció que los reportes de la literatura en cuanto a las modificaciones que se hacen a la base de datos KDD-Cup 99 para el entrenamiento de algoritmos, radica en su mayoría por el tamaño y lo exhaustivo que exige el big data para el entrenamiento de máquinas automáticas en el reconocimiento de patrones.

j. Por lo tanto, para los niveles uno y dos no fue necesaria la caracterización bajo técnicas de extracción de características que condujeran a la generación de un nuevo espacio multivariado de representación. Lo cual se constituye precisamente como un aporte de los clasificadores multinivel, puesto que para el nivel tres, sí fue necesario modificar el espacio de representación bajo rutinas de selección de características, con lo que se logró mayor robustez en la globalidad del esquema estructurado de clasificación de todo el sistema.

5.1. Selección de características

El algoritmo de *Fuzzy Rough Sets* (FRS) arrojó como resultado un sub-conjunto de 10 características, las cuales se pueden observar en la Tabla **5-2**, de las cuales tres características son categorías tipo Básicas, una tipo contenido, dos tipo tráfico y cuatro tipo host, siendo éste último, el adecuado para describir ataques que duran más que el intervalo de las funciones de tiempo estipuladas.

Tabla 5-2.: Selección mediante *Fuzzy Rough Sets*

Características	**Categorías**
Protocol_type	B
Service	B
Wrong_fragment	B
Logged_in	C
Count	T
Srv_serror_rate	T
Dst_host_count	H
Dst_host_svr_count	H
Dst_host_same_src_port_rate	H
Dst_host_same_src_port_rate	H

[1] Básicas (B), Contenido (C), Trafico (T) y Host (H)

De otro modo, *Relief F* entregó como resultado tres sub-conjuntos de 10 características cada uno, como se puede observar en la Tabla **5-3**, las características más relevantes en la clase RUN, fueron tres características en la categoría tipo Básica, dos en la categoría tipo contenido, una en la categoría tipo tráfico y cuatro en la categoría tipo host. Mientras en la forma de ataque de R2L, dio como resultado, una característica que equivale a la categoría tipo Básica, tres son tipo contenido, tres tipo tráfico y tres tipo host, donde la mayor parte de estas categorías están muy homogéneas. Por último en la forma de ataque de U2R se tomaron las tres características que son categorías tipo Básica, tres tipo contenido, una tipo tráfico y tres tipo host,obteniendo un balance en las tres categorías con igual número de características.

5.2. Clasificador Multinivel

El clasificador de 3 niveles propuesto fue probado inicialmente de forma individual en cada nivel realizando la comparación entre tres algoritmos, validando los clasificadores en k-fold con 3 fold debido al alto costo computacional, ya que cada repetición duraba alrededor de de 22 horas, por el gran tamaño y lo exhaustivo que exige el big data para el entrenamiento de

Tabla 5-3.: Selección mediante Relief F

RUN		R2L		U2R	
Características	**Categorías**	**Características**	**Categorías**	**Características**	**Categorías**
Protocol_type	B	Service	B	Service	B
Service	B	Num_failed_logins	C	Dst_bytes	B
Wrong_fragment	B	Logged_in	C	Urgent	B
Logged_in	C	Num_compromised	C	Hot	C
Count	C	Srv_count	T	Logged_in	C
Srv_serror_rate	T	Serror_rate	T	Num_compromised	C
Dst_host_count	H	Srv_serror_rate	T	Srv_count	T
Dst_host_svr_count	H	Dst_host_same_src_port_rate	H	Dst_host_count	H
Dst_host_same_src_port_rate	H	Dst_host_same_src_port_rate	H	Dst_host_same_src_port_rate	H
Dst_host_same_src_port_rate	H	Dst_host_serror_rate	H	Dst_host_svr_serror_rate	H

[1]Básicas (B), Contenido (C), Trafico (T) y Host (H)

máquinas automáticas en el reconocimiento de patrones. La Tabla **5-4**, muestra el desempeño de cada clasificador con porcentaje de precisión para detectar ataques, (verdaderos positivos), donde se describe mediante la ecuación, Precisión= $\frac{VP}{VP+FP}$. Se observa que SVM presenta el mejor desempeño en el Nivel 1 y en el Nivel 2 en los clasificadores de formas de Dos y Probing Attack, con una precisión cercana al 100 %, mientras que k-NN presenta mejor desempeño en el clasificador RUN del Nivel 2 con una precisión del 99 %. Finalmente, SSFCM obtienen el mejor desempeño para los clasificadores de formas de R2L y U2R en el nivel 3, se observa que la precisión más baja se presentó en el clasificador de forma de U2R, el cual no superó el 82 % de precisión.

Tabla 5-4.: Precisión de los clasificadores en porcentaje

Clasificador	**k-NN %**	**SVM %**	**SSFCM %**
Nivel 1 DoS/Pr_Att/RUN	97.394	**99.912**	76.095
Nivel 2 Formas DoS	100	**100**	94.681
Nivel 2 Formas Pr_Att	97.728	**97.810**	90.841
Nivel 2 R2L/U2R/Normal	**99.948**	99.331	0.4554
Nivel 3 Formas R2L	97.195	98.952	**99.190**
Nivel 3 Formas U2R	75.510	78.632	**81.223**

De este modo el clasificador multinivel queda conformado acorde con la (Figura **5-1**), en el cual se aprecia que SVM se usó en el clasificador del Nivel uno y en 2 clasificadores del Nivel dos. k-NN se usó en el clasificador de R2l, U2R y Normal en el nivel tres. Finalmente, SSFCM se usó en los 2 clasificadores de Nivel tres.

Respecto a la selección de características se pueden observar el error global del clasificador por nivel, para cada una de las clases, usando los diferentes métodos de selección de características. Si bien, en la hipótesis se evidencia que la técnica de *Fuzzy Rough Sets* podría aportar una reducción de la dimensionalidad con mayor rendimiento en la clasificación, se pudo evidenciar que la estructura de los datos desde su naturaleza estadística y geométrica,

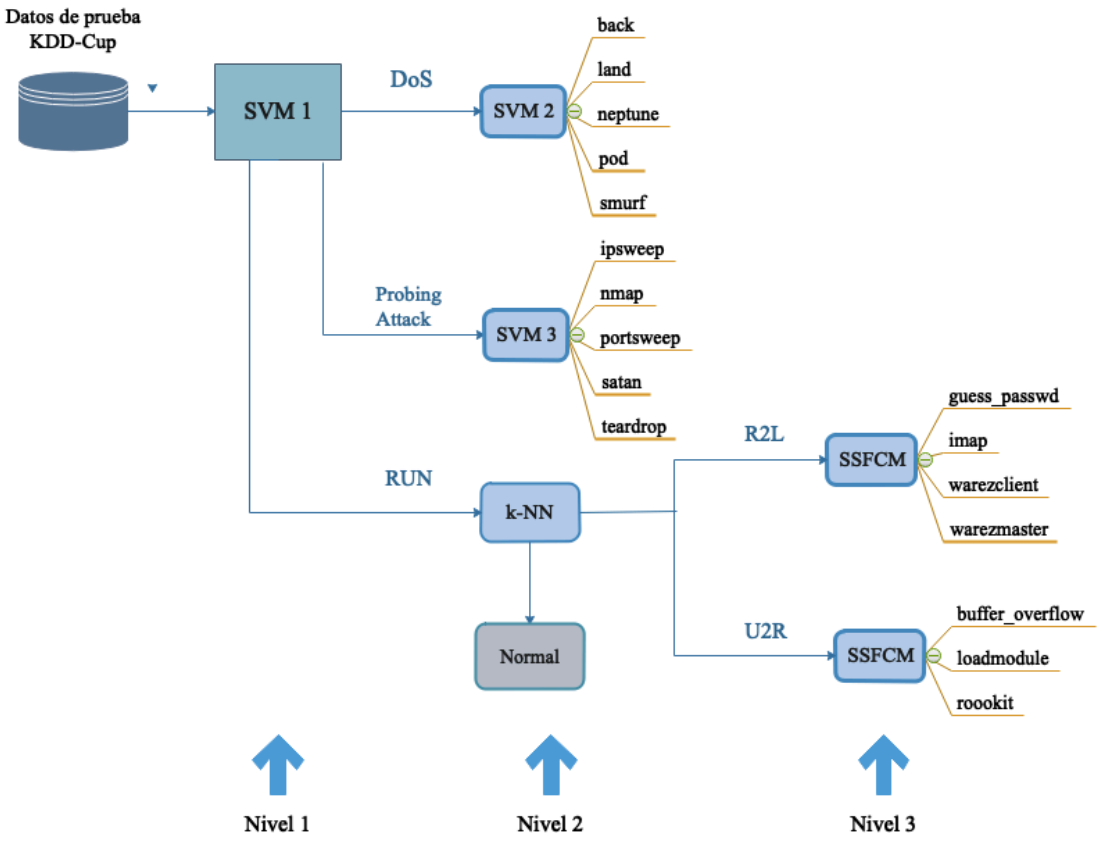


Figura 5-1.: Clasificador Híbrido Multinivel Propuesto

no permitieron que el esfuerzo de la relación transitiva de la similaridad difusa pudiera aproximar adecuadamente las clases de equivalencia. De ésta se manera se puede verificar que el método *Relief F* mejora el desempeño en el nivel 1 como se puede evidenciar en la Tabla **5-5**.

Tabla 5-5.: Error general del clasificador nivel 1

Clasificador	Clases	Sin selección	FRS	*Relief F*
Nivel 1 DoS/PA/RUN	DoS	0,0038	0,0038	**0,0037**
	PA	0,0048	0,0046	0,0046
	RUN	0,0042	0,0043	0,0042

En el nivel 2, se obtuvo cuatro clases con el error más bajo, cinco clases con el error igual al clasificador sin selección y con FRS, y las otras cuatro para el clasificador sin selección y FRS, ver Tabla **5-6**.

Por último en el nivel tres, obtuvo una clase con el error mas bajo, ver Tabla **5-7**.

Tabla 5-6.: Error general del clasificador nivel 2

Clasificador	Clases	Sin selección	FRS	*Relief f*
Nivel 2 Formas DoS	back	0,0009	**0,0007**	0,0008
	land	0,0001	0,0001	0,0001
	neptune	0,0002	0,0002	0,0002
	pod	0,0001	0,0001	0,0001
	smurf	0,0034	0,0035	0,0034
Nivel 2 Formas PA	ipsweep	0,3374	0,3398	**0,3232**
	nmap	0,1402	0,1331	**0,1283**
	portsweep	**0,0499**	0,0523	0,0523
	satan	0,2970	0,2852	**0,2543**
	teardrop	0,0000	0,0000	0,0000
Nivel 2 R2L/U2R/Normal}	R2L	0,0056	**0,0007**	0,0043
	U2R	0,0031	0,0014	**0,0011**
	Normal	**0,0198**	1,0177	1,8327

Esto sugiere que usar *Relief F* mejora el desempeño del clasificador, por la mayor cantidad de clases con el error global mas bajo, no obstante se realizó el test estadístico no paramétrico de Friedman, para corroborar la significancia estadística de la mejoría, encontrando un valor p de 0.02, lo que muestra una diferencia estadística significativa entre las estrategias de selección de características, el ranking de medias para el error general de las estrategias Sin selección, FRS y *Relief F* del test fue de 2.3478 %, 2.0652 % y 1.5870 % , respectivamente. Confirmando que el clasificador multinivel con *Relief F* ofrece el mejor desempeño.

Tabla 5-7.: Error general del clasificador nivel 3

Clasificador	Clases	Sin selección	FRS	*Relief F*
Nivel 3 Formas R2L	guess_passwd	0,3620	**0,0905**	0,1810
	imap	0,2715	**0,0000**	0,3620
	warezclient	4,0724	32,3982	**0,1810**
	warezmaster	0,3620	0,1810	**0,1006**
Nivel 3 Formas U2R	buffer_overflow	34,6939	34,2857	**5,8824**
	loadmodule	16,3265	11,4286	**5,8824**
	rootkit	20,4082	**2,8571**	7,3529

Pese a que la selección de características con *Relief F* aumenta en mas 1 unidad porcentual el error en la clasificación de la clase normal, se decidió mantener la selección dada la mejoría en la detección de ataques R2L y U2R. De otro modo, se observa que *Relief F* disminuye considerablemente el error en la detección de formas de U2R.

En la Tabla **5-8** se observa el número de muestras que pertenecían a alguna forma de ataque pero fueron clasificadas como normales (Falsos negativos), al igual que las muestras que pertenecía a la clase normal y fueron clasificados en alguna forma de ataque (Falsos positivos). Se puede observar que el total de falsos negativos es de 348 en el clasificador con *Relief F*, lo que representa solo un 0.007 % de la totalidad de muestras, y para el clasificador sin selección fue de 497 lo que representa el 0,010 %, esto muestra la robustez del algoritmo para detectar los ataques en sus diversas formas, tener un bajo numero de falsos negativos es de vital importancia, ya que éstos son los ataques que pasan desapercibidos por el IDS. De otro modo, el número de falsos positivos es de 17918, representando un 0.37 % de la totalidad de muestras, mientras que para el clasificador sin selección reporto una cantidad muy pequeña de falsos positivos para un total de 263 lo que corresponde a un 0,0053 %, los falsos positivos generan alarmas en el IDS, pero no corresponden a verdaderos ataques, se observa que las formas de R2L y U2R, en el clasificador con *Relief F*, son aquellas que más muestras aportan a los falsos positivos, mostrando así que la mayor debilidad del algoritmo se encuentra en el tercer clasificador del nivel 2, es decir en el clasificador entre R2L, U2R y Normal, que fue realizado con k-NN, esto se puede ocasionar debido al desbalance de clases, la falta de información de las características o la similitud entre las características de estos dos ataques con las conexiones Normales.
En este trabajo, se realizaron diferentes pruebas que permitieran incrementar la robustez en la clasificación del nivel 3, lo cual se consiguió realizando selección de características en el nivel, indicando que existen características redundantes o irrelevantes para las diferentes formas de ataque identificadas en el primer y segundo nivel, se convirtieron en relevantes para la robustez de la clasificación del tercer nivel como se puede ver en la tabla **5-9** con un % total, de (Falsos Positivos) inferior al 1 %. El 98.17 % de los datos normales fueron clasificados correctamente (Verdaderos Negativos) y el 99.68 % de los ataques fueron clasificados correctamente en su forma (Verdaderos Positivos). Finalmente, sólo 226 muestras (0.004 % de la base de datos) de alguna forma de ataque fueron clasificadas erróneamente en otra forma de ataque diferente .

5.3. Discusión

La discusión de esta tesis, se presenta mediante la comparación de los resultados del esquema propuesto en este trabajo con los resultados hallados en la literatura que hasta el momento son los de mayor desempeño encontrados.
En [104], propusieron un clasificador híbrido multinivel que usa diferentes conjuntos de ca-

Tabla 5-8.: Número de ataques clasificados como normal y normales clasificados como ataque

	Clasificador con *Relief F*		**Clasificador sin selección**	
Forma de ataque	**Ataques clasificados como normales**	**Normales clasificados como ataques**	**Ataques clasificados como normales**	**Normales clasificados como ataques**
back	28	87	33	87
land	1	7	6	7
neptune	3	4	8	3
pod	0	0	5	0
smurf	126	4	132	3
ipsweep	73	69	81	72
nmap	21	17	27	17
portsweep	14	7	18	9
satan	74	8	108	5
teardrop	0	0	0	0
guess_passwd	0	4938	4	1
imap	0	79	2	3
warezclient	4	9041	45	37
warezmaster	1	871	3	2
buffer_overflow	0	721	10	4
loadmodule	0	641	6	3
rootkit	3	1424	9	10
Total	**348**	17918	497	**263**

Tabla 5-9.: Desempeño Global Del clasificador multinivel

Desempeño Global	**VP**	**VN**	**FP**	**FN**
Clasificador multinivel propuesto	99.68 %	98.17 %	0.315 %	0.0053 %

racterísticas sobre cada clasificador, utilizan el método Función de discernibilidad basada en selección de características (DFBFS, por sus siglas en inglés), como un selector basado en wrapper en la primera etapa. Para la detección de intrusos en KDD cup 99. usaron el 10 % del conjunto de entrenamiento y para la prueba el KDD test en un proceso de selección aleatoria del IDS, donde utilizaron un clasificador de tres niveles así: en el nivel uno quedó con las clases DoS y PA, en el segundo nivel con R2L y U2R y en el último nivel con Normal, ellos utilizaron un clasificador por nivel, mientras que se utilizó un clasificador Híbrido de tres niveles para tener un mejor desempeño debido a la construcción de los clasificadores en cada nivel, dispuestos así: uno en el primer nivel, tres en el nivel dos y en el nivel tres, se contó con dos para darle mejor precisión a las clases minoritarias y que son muy semejantes, debido a eso se optó por seleccionar las mejores características en el nivel donde estaba la clase RUN para que el clasificador pueda detectar las formas de ataque de cada clase ver Tabla **5-10**.

En [165], utilizaron como método de selección de características basado en Wrapper, usaron una función de disernibilidad para lo cual discretizaron los datos con una medida de entro-

pía, ocasionando pérdida de información.En este trabajo de investigación se utilizaron los métodos de selección de características FRS y *Relief F*, que demostró que el segundo obtuvo mejor selección, que opera tanto con datos continuos como discretos, por esta razón no es muy susceptible a la pérdida de información. Ellos utilizaron el 10 % de base de datos KDD Cup 99, mientras aquí se utilizó la base de datos completa. El clasificador que implementaron se describe así: para el primer nivel la clase R2L con el método Back-Propagation, que clasificaba esa clase de ataque y si no es la pasa a la siguiente clase, así sucesivamente hasta llegar a normal, porque en los otros niveles utilizaron el clasificador C4.5, en cambio aquí se empleó en los tres niveles antes descritos, lo cual nos dio mejores resultados exceptuando en DoS que ambos nos arrojó un resultado igual.

En [9], trabajaron con el 10 % de la KDD Cup 99 y con la KDD Cup corregida, cabe anotar que este trabajo se realizó con toda la base completa. se implementó un enfoque híbrido de dos niveles, de lo cual en el primer nivel utilizan un método de *clustering* y en la etapa dos utilizan k-NN para determinar si es un ataque o Normal para tener alto índice de precisión del sistema y alta tasa de falsos positivos, entretanto el clasificador propuesto por nosostros tiene un mayor desempeño en el porcentaje global reportado en Tabla **5-10**.

En [163], prepusieron un modelo híbrido de dos niveles, donde en la etapa uno implementan un cluster jerárquico de las clases ataque, que denominan C1 a C4 y la clase Normal, ya en la segunda etapa contiene 2 componentes, donde el primero procesa las diferentes predicciones de probabilidad. El segundo es el clasificador final que decide si la conexión es un comportamiento normal o un ataque. para el sistema utilizan el 10 % de la KDD Cup 99 y la KDD Cup Corregida, obteniendo mejores resultandos en U2R y Normal, mientras que nuestro modelo alcanzo mejores resultados en los otros resultados de la Tabla **5-10**, debido a la buena implementación en el nivel donde se selecciono con el método de *Relief F*, para encontrar las mejores características de la clase RUN y así poder clasificar con mejor desempeño.

Se puede evidenciar la integración de los resultados en las matrices de confusión del clasificador propuesto, con las diferentes configuraciones en el anexo Figura **A-1**, **B-1** y **C-1**

Tabla 5-10.: Comparación Métodos

MÉTODO HÍBRIDO	DoS	Probing_Attack	R2L	U2R	Normal	Precisión	T_D	T_FA
Método propuesto	**99.99**	**99.56**	**96.19**	77.55	98.16	**99.69**	**98.62**	**0.36**
SVM y ELM basado en k-means modificado (al-yaseen2017)(1)	99.54	87.22	31.39	21.93	98.13	95.17	95.17	1.87
MHCVF (akyol2016)(2)	**99.99**	99.39	84.04	80	94.29	98.03	NR	5.71
Enfoque híbrido de dos niveles (guo2016)(3)	97.32	98.34	12.85	79.38	**99.22**	93.29	91.86	0.78
HCPTC-IDS (ahmim2018)(4)	99.83	95.27	36.50	**81.14**	98.87	96.27	95.65	1.13

6. Conclusiones

Se propuso un esquema de clasificación híbrido multinivel para lograr consistencia en cuanto a precisión en un sistema de detección de intrusos para cuatro tipos de ataque: *Denial of Service (DoS), Probing Attack (PA), Remote to Local (R2L), User to Root (U2R)* en redes computacionales. La metodología propuesta incluyó diversas pruebas asociadas a configuraciones con *Fuzzy Rough Sets* y *Relief F* en la parte de la construcción del espacio efectivo de representación multivariada; y tomando en cuenta la robustez para la identificación de los ataques, se propusieron tres niveles de clasificación compuesto por seis clasificadores de tres tipos: Máquinas de Vectores de Soporte (SVM), k- Vecinos más Cercanos (k-NN) y *Fuzzy c-Means* Semi-Supervisado (SSFCM), donde cada nivel quedó estructurado así:

- *Nivel 1:* Un primer clasificador identifica dos clases y una agrupación: DoS, PA y la agrupación compuesta por las clases R2L, U2R y Normal.
- *Nivel 2:* Compuesto por tres clasificadores: los dos primeros identifican las formas de ataque correspondientes a DoS y PA, respectivamente, mientras el tercer clasificador identifica las clases R2L, U2R y Normal.
- *Nivel 3:* Donde dos clasificadores identifican las formas de ataque para R2L y U2R.

De acuerdo con el primer objetivo específico de la tesis, se puede concluir que la generación del espacio de representación multivariada que logró mayor efectividad en cuanto a precisión en la identificación de ataques informáticos fue la obtenida con la selección de características mediante la técnica *Relief F*, según los resultados presentados en las Tablas **5-5**, **5-6** y **5-7**, en donde se usó el test estadístico no paramétrico de Friedman. Si bien, en la hipótesis se supuso que la técnica de *Fuzzy Rough Sets* podría aportar una reducción de la dimensionalidad con mayor impacto en la clasificación, se evidenció que la estructura de los datos desde su naturaleza estadística y geométrica, no permitieron que el esfuerzo de la relación transitiva de la similaridad difusa pudiera aproximar adecuadamente las clases de equivalencia. Sin embargo, aunque *Relief F* no se ofreció para el primer y segundo nivel un espacio de representación reducido que derivó en resultados de clasificación con altos niveles de precisión y robustez en cuanto a falsos positivos, falsos negativos y verdaderos negativos, para el tercer nivel decayó la robustez en la clasificación en cuanto a falsos positivos. Este hallazgo, también había sido enunciado en otros trabajos reportados en la literatura, por ejemplo en: [163], [104], [165] y [9]. En esta tesis, se hicieron diferentes pruebas que permitieran incrementar la robustez en la clasificación del nivel 3, lo cual se consiguió implementando el proceso de selección de

características, por lo que se pudo evidenciar que las características redundantes o irrelevantes disminuían en el espacio de decisión la variabilidad entre-clases en cuanto a las formas de ataque, siendo el hallazgo de las características relevantes imprescindible para generar un nuevo espacio de representación reducido más robusto para la clasificación del tercer nivel.

Tomando en cuenta el segundo objetivo específico de la tesis, se puede concluir que las configuraciones de clasificación híbrida multinivel fueron obtenidas comparando el desempeño en cada nivel de tres tipos de clasificadores: SVM, k-NN y SSFCM (ver Figura **5-1**), donde se evidenció de manera general que SVM presentó el mejor desempeño para los niveles 1 y 2 con precisiones de clasificación cercanas al 100 % (ver Tabla **5-4**). En cuanto al tercer nivel, se encontró que la naturaleza de los datos correspondientes a las clases R2L y U2R, eran muy similares entre sí y con muy baja variabilidad respecto de la clase normal, incrementando el riesgo de que estos ataques se confundieran dentro de lo considerado como un tráfico sin amenazas. Es aquí donde el clasificador SSFCM aportó desde sus agrupamientos a una solución significativa, dado el poder de flexibilidad que le permite la operación difusa en las asignaciones de pertenencia, por lo que este clasificador aportó los mejores resultados en precisión y robustez para el nivel 3, como se puede revisar en la Tabla **5-4**.

Considerando el tercer objetivo específico de la tesis, se puede concluir que para validar el esquema de clasificación híbrido multinivel a un grado de detalle que cubriera la identificación de las diferentes formas de ataque (correspondientes a DoS, PA, R2L y U2R), era necesario el uso de la base de datos completa, aunque en la literatura no son frecuentes los reportes con resultados experimentales sobre toda esta base de datos por su gran tamaño y la exigencia computacional que esto conlleva. En este sentido, la experimentación compuesta por el entrenamiento y validación de los algoritmos en una base de datos tan extensa (i.e., 4.898.430 datos de 41 dimensiones), exigió un preprocesamiento de datos mediante rutinas de análisis estadístico multivariado y optimización de los algoritmos, para que las pruebas tuvieran niveles de consistencia, confiabilidad y significancia. Revisando las Tablas **5-5**, **5-6** y **5-7**, se puede evidenciar que para el primer nivel se lograron tasas de acierto en la clasificación cercanas al 100 %, mientras que para el nivel 2 fueron cercanas al 98 %. En cuanto al nivel 3, que para la literatura es uno de los retos interesantes de identificación automática, se lograron tasas de acierto alrededor del 92 %. Estos resultados de validación logran un nivel de importancia, al tomarse en cuenta que fueron logradas sobre la base de datos completa, superando en la mayoría de los casos los resultados reportados hasta el momento en la comunidad científica, y tomando en cuenta los reportes de la literatura donde se presentan resultados comparables con los de esta tesis, éstos fueron obtenidos usando solamente el 10 % de los datos o sobre versiones de la base de datos reducida por correcciones que disminuyen complejidad en la estructura de la información.

Como trabajo futuro, queda pendiente la propuesta de nuevos desarrollos experimentales

para entrenar y validar el sistema propuesto en esta tesis, usando otras bases de datos que contengan nuevas formas de ataque o conjuntos de datos dinámicos con captura de las composiciones e intervenciones de tráfico, como pueden ser las bases de datos: Kyoto-2006, CICIDS2017 y la CSE-CIC-IDS2018. También el modelo propuesto puede ampliarse para implementar Sistemas de Detección de Intrusos (IDS) con detección en línea, enfocándose a una exploración más exhaustiva de la interoperabilidad, nuevos enfoques de toma de decisión y rutinas de aprendizaje automático. Asimismo, se pueden orientar mayores esfuerzos investigativos al logro de un IDS bajo entornos virtualizados. Esta área de trabajo ha ganado preponderancia e importancia en los entornos corporativos y, por ende, la operabilidad en la nube debe estar lo suficientemente protegida contra cualquier tipo de amenazas. Así, entonces, se puede continuar el trabajo de esta tesis en la articulación informática del IDS propuesto con las nuevas necesidades en el campo de la Ciberseguridad.

A. Anexo: Matriz de confusión sin selección

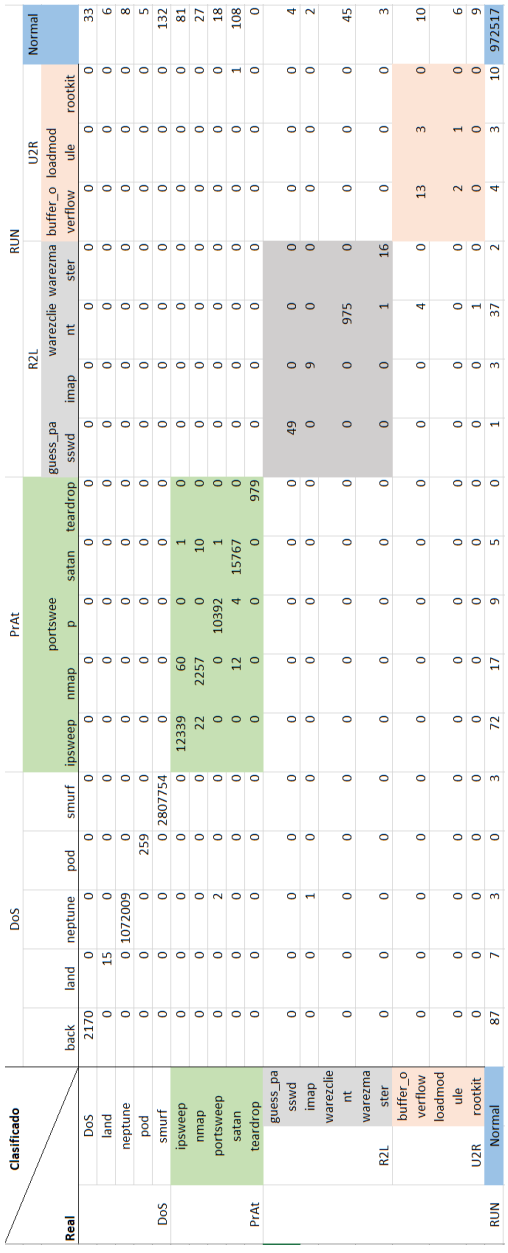


Figura A-1.: Matriz de confusión sin selección

B. Anexo: Matriz de confusión con FRS

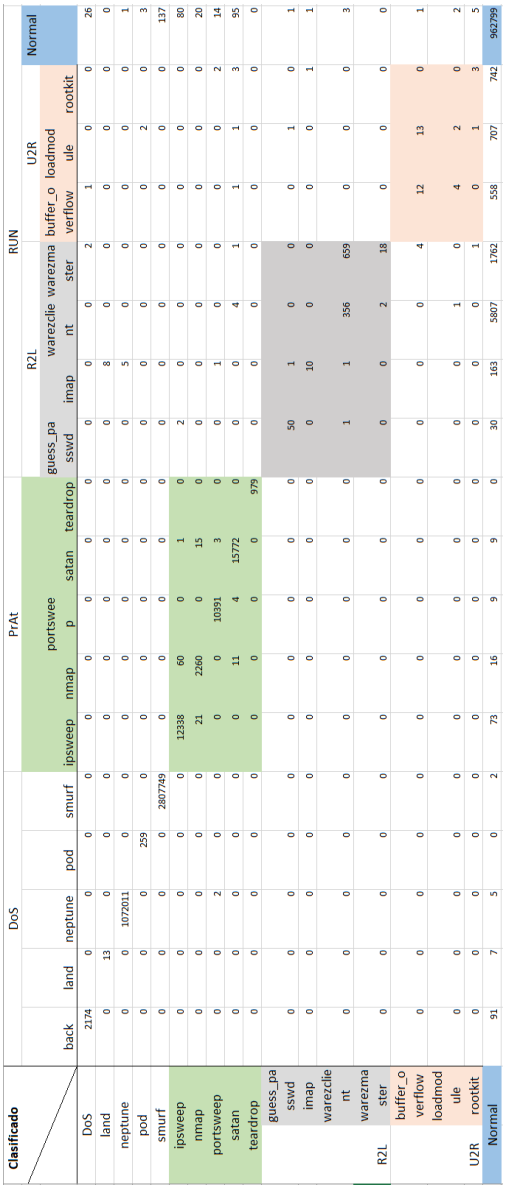


Figura B-1.: Matriz de confusión con FRS

C. Anexo: Matriz de confusión con *Relief F*

Real \ Clasificado		DoS					PrAt					RUN					U2R		Normal
		back	land	neptune	pod	smurf	ipsweep	nmap	portsweep	satan	teardrop	guess_passwd	imap	warezclient	warezmaster	buffer_overflow	loadmodule	rootkit	
DoS	DoS	2173	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	28
	land	0	15	0	0	0	0	0	0	0	0	0	1	0	1	1	2	0	1
	neptune	0	0	107201 1	0	0	0	0	0	0	0	0	1	0	1	0	1	0	3
	pod	0	0	0	259	0	0	0	0	0	0	0	0	0	2	1	2	0	0
	smurf	0	0	0	0	2807753	0	0	0	0	0	7	0	0	0	0	0	0	126
PrAt	ipsweep	0	0	0	0	0	12341			0	0	1	0	0	2	2	2	0	73
	nmap	0	0	0	0	0	21			0	0	0	0	0	0	0	0	0	21
	portsweep	0	0	1	0	0	0		10393		0	0	0	0	0	0	0	3	14
	satan	0	0	0	0	0	0		3		0	19	0	5	0	0	2	4	74
	teardrop	0	0	0	0	0	0		0		979	0	0	0	0	0	0	0	0
R2L	guess_passwd	0	0	0	0	0	0	0	0	0	0	48	0	1	1	1	2	0	0
	imap	0	0	2	0	0	0	0	0	0	0	1	4	0	4	0	0	1	0
	warezclient	0	0	0	0	0	0	0	0	0	0	0	3	983	0	30	0	0	4
	warezmaster	0	0	0	0	0	0	0	0	0	0	0	0	0	16	1	1	1	1
	buffer_overflow	0	0	0	0	0	0	0	0	0	0	0	0	5	0	22	3	0	0
U2R	loadmodule	0	0	0	0	0	0	0	0	0	0	0	0	1	0	3	5	0	0
	rootkit	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	5	3
RUN	Normal	87	7	4	0	4	69	17	7	8	0	4938	79	9041	871	721	641	1424	954862

Figura C-1.: Matriz de confusión con *Relief F*

Bibliografía

[1] Rpdo Dqgkl, E H Dffhvvhg, Qwlyluxv Vriwzduh, R U Lqwuxvlrq, Ghwhfwlrq V Vwhpv, W K H Qhwzrun, Wudiilf Xvlqj, W K H Ydulrxv, Qhwzrun Dqdo, and Vlqj Wrrov. 1hwzrun 6hfxulw\ 3ureohpv dqg 6hfxulw\ $wwdfnv. pages 3855–3857, 2016.

[2] Abdulghani Ali Ahmed, Noorul Ahlami, and Kamarul Zaman. Attack Intention Recognition : A Review. 19(2):244–250, 2017.

[3] Tarfa Hamed, Jason B Ernst, and Stefan C Kremer. A Survey and Taxonomy of Classifiers of Intrusion Detection Systems. 2018.

[4] Symantec Corporation. ISTR Internet Security Threat Report Volume 24 | 02/19. 24(February), 2019.

[5] Q Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. Journal of Network and Computer Applications. *Journal of Network and Computer Applications*, pages 1–13, 2015.

[6] Mark Stamp. A Survey of Machine Learning Algorithms and Their Application in Information Security. pages 33–55. Springer International Publishing, 2018.

[7] Preeti Mishra, Vijay Varadharajan, Uday Tupakula, and Emmanuel S. Pilli. A detailed investigation and analysis of using machine learning techniques for intrusion detection. *IEEE Communications Surveys and Tutorials*, 21(1):686–728, 2019.

[8] Chibuzor John Ugochukwu and E O Bennett. An Intrusion Detection System Using Machine Learning Algorithm. *International Journal of Computer Science and Mathematical Theory*, 4(1):39–47, 2018.

[9] Chun Guo, Yuan Ping, Nian Liu, and Shou Shan Luo. A two-level hybrid approach for intrusion detection. *Neurocomputing*, 214:391–400, 2016.

[10] Atilla Özgür and Hamit Erdem. A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015. *PeerJ PrePrints*, 4:1–21, 2016.

[11] Amol M.Pawar and Manisha S. Mahindrakar. A Comprehensive Survey on Online Anomaly Detection. *International Journal of Computer Applications*, 119(17):41–45, 2015.

[12] Pratap Dangeti. *Statistics for Machine Learning Build supervised, unsupervised, and reinforcement learning models using both Python and R.* 2017.

[13] Atulya Nagar, Durga Prasad Mohapatra, and Nabendu Chaki. Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics: ICACNI 2015, Volume 1. *Smart Innovation, Systems and Technologies*, 43, 2016.

[14] Unnati R Raval and Chaita Jani. Implementing and Improvisation of K-means Clustering. *International Journal of Computer Science and Mobile Computing*, 4(11):72–76, 2015.

[15] Can Jin, Zhiwei Ye, Chunzhi Wang, Lingyu Yan, and Ruoxi Wang. A network intrusion detection method based on hybrid rice optimization algorithm improved fuzzy c-means. *Proceedings of the 2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS-SWS 2018*, pages 47–52, 2018.

[16] Shikha Agrawal and Jitendra Agrawal. Survey on Anomaly Detection using Data Mining Techniques. *Procedia Computer Science*, 60(1):708–713, 2015.

[17] Arjunwadkar Narayan M. An Intrusion Detection System , (IDS) with Machine Learning (ML) Model Combining Hybrid Classifiers. 2(4):647–651, 2015.

[18] V. B. Surya Prasath, Haneen Arafat Abu Alfeilat, Omar Lasassmeh, and Ahmad B. A. Hassanat. Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbor Classifier - A Review. 2019.

[19] Elike Hodo, Xavier Bellekens, Andrew Hamilton, Christos Tachtatzis, and Robert Atkinson. Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey. pages 1–43, jan 2017.

[20] Manish Verma and Sanjay Kumar Jena. *A Multi-Stage Intrusion Detection Approach for Network Security.* PhD thesis, 2015.

[21] Sharmila KishorWagh, Vinod K. Pachghare, and Satish R. Kolhe. Survey on Intrusion Detection System using Machine Learning Techniques. *International Journal of Computer Applications*, 78(16):30–37, 2013.

[22] Teresa F Lunt, R Jagannathan, Rosanna Lee, Alan Whitehurst, and Menlo Park. Knowledge-Based Intrusion Detection. 1989.

[23] Sheenam Sheenam and Sanjeev Dhiman. Comprehensive Review: Intrusion Detection System and Techniques. *IOSR Journal of Computer Engineering*, 18(04):20–25, 2016.

[24] M Azhagiri, A Rajesh, and S Karthik. Intrusion Detection and Prevention System: Issues and Challenges. *{International Journal of Applied Engineering Research*, 10(87):1–12, 2015.

[25] Jubeen Shah. Understanding and study of intrusion detection systems for various networks and domains. *2017 International Conference on Computer Communication and Informatics, ICCCI 2017*, pages 1–6, 2017.

[26] Hanan Hindy, David Brosset, Ethan Bayne, Amar Seeam, Christos Tachtatzis, Robert Atkinson, and Xavier Bellekens. A Taxonomy and Survey of Intrusion Detection System Design Techniques, Network Threats and Datasets. 1(1), 2018.

[27] Ahmed AlEroud and George Karabatis. Methods and techniques to identify security incidents using domain knowledge and contextual information. *Proceedings of the IM 2017 - 2017 IFIP/IEEE International Symposium on Integrated Network and Service Management*, pages 1040–1045, 2017.

[28] Felix Erlacher and Falko Dressler. FIXIDS: A high-speed signature-based flow intrusion detection system. *IEEE/IFIP Network Operations and Management Symposium: Cognitive Management in a Cyber World, NOMS 2018*, pages 1–8, 2018.

[29] Raihan Ul Islam, Mohammad Shahadat Hossain, and Karl Andersson. A novel anomaly detection algorithm for sensor data under uncertainty. *Soft Computing*, 22(5):1623–1639, mar 2018.

[30] Santhosh Kelathodi Kumaran, Debi Prosad Dogra, and Partha Pratim Roy. Anomaly Detection in Road Traffic Using Visual Surveillance: A Survey. pages 1–14, jan 2019.

[31] Seçil Taburoğlu. A Survey on Anomaly Detection and Diagnosis Problem in the Space System Operation, 2019.

[32] Asaf Shabtai, Uri Kanonov, and Yuval Elovici. Intrusion detection for mobile devices using the knowledge-based, temporal abstraction method. *Journal of Systems and Software*, 83(8):1524–1537, 2010.

[33] Shijoe Jose, D. Malathi, Bharath Reddy, and Dorathi Jayaseeli. A Survey on Anomaly Based Host Intrusion Detection System. *Journal of Physics: Conference Series*, 1000(1):0–10, 2018.

[34] Vahid Kaviani Jabali. Taxonomy of Feature selection in Intrusion Detection System. *IJCSNS International Journal of Computer Science and Network Security*, 17(6):88–102, 2017.

[35] Raj Jain. Metrics , Techniques and Tools of Anomaly Detection : A Survey. pages 1–12, 2017.

[36] W. Liao, B. Rosenhahn, and M. Ying Yang. Gaussian Process for Activity Modeling and Anomaly Detection. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3/W5:467–474, 2015.

[37] Jingjing Fei and Shiliang Sun. *Online Anomaly Detection with Sparse Gaussian Processes*. Number May 2019. 2019.

[38] Nurudeen Mahmud and Anazida Zainal. Intrusion Detection Techniques in Cloud Computing: A Review. *International Journal of Computer Applications*, 179(12):26–33, 2018.

[39] Gianluigi Folino and Pietro Sabatino. Ensemble based collaborative and distributed intrusion detection systems: A survey. *Journal of Network and Computer Applications*, 66:1–16, 2016.

[40] Chia Mei Chen, Dah Jyh Guan, Yu Zhi Huang, and Ya Hui Ou. Anomaly network intrusion detection using Hidden Markov Model. *International Journal of Innovative Computing, Information and Control*, 12(2):569–580, 2016.

[41] Pilar Holgado, VICTOR A. VILLAGRA, and Luis Vazquez. Real-time multistep attack prediction based on Hidden Markov Models. *IEEE Transactions on Dependable and Secure Computing*, 5971(c), 2017.

[42] Md Reazul, Abdur Rahman, and Tanvir Samad. A Network Intrusion Detection Framework based on Bayesian Network using Wrapper Approach. *International Journal of Computer Applications*, 166(4):13–17, 2017.

[43] Yu-Yang Zhou and Guang Cheng. An Efficient Network Intrusion Detection System Based on Feature Selection and Ensemble Classifier. 2019.

[44] Rupam Kr. Sharma, Hemanta Kumar Kalita, and Parashjyoti Borah. Analysis of Machine Learning Techniques Based Intrusion Detection Systems. In *Smart Innovation, Systems and Technologies*, volume 43, pages 485–493. 2016.

[45] Asghar AliShah, Malik Sikander Hayat Khiyal, and Muhammad Daud Awan. Analysis of Machine Learning Techniques for Intrusion Detection System: A Review. *International Journal of Computer Applications*, 119(3):19–29, jun 2015.

[46] J. R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1:81–106, 1986.

[47] J Ross, Quinlan Morgan Kaufmann, and Steven L Salzberg. Book Review: C4.5: Programs for Machine Learning. *Machine Learning*, 240:1–6, 1994.

[48] Adel Sabry Eesa, Zeynep Orman, and Adnan Mohsin Abdulazeez Brifcani. A new feature selection model based on ID3 and bees algorithm for intrusion detection system. *Turkish Journal of Electrical Engineering and Computer Sciences*, 23(2):615–622, 2015.

[49] Sachin Prakash Gavhane & Vijay Maruti Shelake. Intrusion Detection System Using Optimal C4.5 Algorithm. *International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR)*, 4(2):5–14, 2014.

[50] F. Lydia Catherine, Ravi Pathak, and V. Vaidehi. Efficient host based intrusion detection system using Partial Decision Tree and Correlation feature selection algorithm. *2014 International Conference on Recent Trends in Information Technology, ICRTIT 2014*, pages 0–5, 2014.

[51] Kai Peng, Victor C.M. Leung, Lixin Zheng, Shangguang Wang, Chao Huang, and Tao Lin. Intrusion detection system based on decision tree over big data in fog environment. *Wireless Communications and Mobile Computing*, 2018, 2018.

[52] Srinivas Mukkamala, Andrew H. Sung, and Ajith Abraham. Intrusion detection using an ensemble of intelligent paradigms. *Journal of Network and Computer Applications*, 28(2):167–182, 2005.

[53] D KS and BB Ramakrishna. An Artificial Neural Network based Intrusion Detection System and Classification of Attacks. *International Journal of Engineering Research and Applications*, 3(4):1959–1964, 2014.

[54] Akashdeep, Ishfaq Manzoor, and Neeraj Kumar. A feature reduced intrusion detection system using ANN classifier. *Expert Systems with Applications*, 88:249–257, 2017.

[55] L. A Zadeh. Fuzzy Sets. *Journal of Plant Pathology*, (8):338–353, 1965.

[56] Timothy A. Bonin, Brian J. Carroll, R. Michael Hardesty, W. Alan Brewer, Kristian Hajny, Olivia E. Salmon, and Paul B. Shepson. Doppler lidar observations of the mixing height in Indianapolis using an automated composite fuzzy logic approach. *Journal of Atmospheric and Oceanic Technology*, 35(3):473–490, 2018.

[57] Nenekazi Nokuthala Penelope Mkuzangwe and Fulufhelo Vincent Nelwamondo. A Fuzzy Logic Based Network Intrusion Detection System for Predicting the TCP SYN Flooding Attack. In Ngoc Thanh Nguyen, Satoshi Tojo, Le Minh Nguyen, and Bogdan Trawiński, editors, *Journal of Intelligent and Fuzzy Systems*, volume 10192 of *Lecture Notes in Computer Science*, pages 14–22. Springer International Publishing, Cham, 2017.

[58] Klawonn F Kruse R Nürnberger A Michels K. *Fuzzy control: Fundamentals, stability and design of fuzzy controllers*, volume 200. 2006.

[59] Suresh Chandra Satapathy, Amit Joshi, Nilesh Modi, and Nisarg Pathak. Proceedings of international conference on ICT for sustainable development: ICT4SD 2015 volume 1. *Advances in Intelligent Systems and Computing*, 408:809–815, 2016.

[60] Nenekazi Nokuthala, Penelope Mkuzangwe, and Fulufhelo Nelwamondo. *A Fuzzy Logic Based Network Intrusion Detection System for Predicting the TCP SYN Flooding Attack*, volume 10192 of *Lecture Notes in Computer Science*. Springer International Publishing, Cham, 2017.

[61] R Shanmugavadivu and DN N Nagarajan. Network intrusion detection system using fuzzy logic. *Indian Journal of Computer Science and Engineering*, 2(1):101–111, 2011.

[62] Al Mehedihasan, Mohammed Nasser, and Biprodip Pal. On t he KDD ' 99 Dataset : Support Vector Machine Based Intrusion Detection System (IDS) with Different Kernels. *Ijecce*, 4(4):1164–1170, 2013.

[63] Helmi B.Md Rais and Tahir Mehmood. Feature selection in intrusion detection, state of the art: A review. *Journal of Theoretical and Applied Information Technology*, 94(1):30–43, 2016.

[64] Clarence Chio and David Freeman. *Machine Learning and Security*. O'Reilly Media, Inc, first edit edition, 2018.

[65] Kamran Siddique, Zahid Akhtar, Farrukh Aslam Khan, and Yangwoo Kim. KDD Cup 99 Data Sets: A Perspective on the Role of Data Sets in Network Intrusion Detection Research. *Computer*, 52(2):41–51, 2019.

[66] Suad Mohammed Othman, Fadl Mutaher Ba-Alwi, Nabeel T. Alsohybe, and Amal Y. Al-Hashida. Intrusion detection model using machine learning algorithm on Big Data environment. *Journal of Big Data*, 5(1), 2018.

[67] Huiwen Wang, Jie Gu, and Shanshan Wang. An effective intrusion detection framework based on SVM with feature augmentation. *Knowledge-Based Systems*, 136:130–139, 2017.

[68] Jie Gu, Lihong Wang, Huiwen Wang, and Shanshan Wang. A novel approach to intrusion detection using SVM ensemble with feature augmentation. *Computers and Security*, 86:53–62, 2019.

[69] B. W. Silverman and M. C. Jones. E. Fix and J.L. Hodges (1951): An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation. *International Statistical Review*, 57(3):233–238, 1989.

[70] Vitoantonio Bevilacqua and Atul Negi. *Recent Trends in Image Processing and Pattern Recognition*, volume 709. 2017.

[71] Zhiwen Yu, Hantao Chen, Jiming Liuxs, Jane You, Hareton Leung, and Guoqiang Han. Hybrid κ-Nearest Neighbor Classifier. *IEEE Transactions on Cybernetics*, 46(6):1263–1275, 2016.

[72] Zhenghui Ma and Ata Kaban. K-Nearest-Neighbours with a novel similarity measure for intrusion detection. *2013 13th UK Workshop on Computational Intelligence, UKCI 2013*, pages 266–271, 2013.

[73] Aboosaleh M Sharifi, Saeed K Amirgholipour, and Alireza Pourebrahimi. Intrusion Detection Based on Joint of K-Means and KNN. *Journal of Convergence Information Technology*, 10(5):42, 2015.

[74] Shweta Malhotra, Vikram Bali, and K. K. Paliwal. Genetic programming and K-nearest neighbour classifier based intrusion detection model. *Proceedings of the 7th International Conference Confluence 2017 on Cloud Computing, Data Science and Engineering*, pages 42–46, 2017.

[75] Wenchao Li, Ping Yi, Yue Wu, Li Pan, and Jianhua Li. A New Intrusion Detection System Based on KNN Classification Algorithm in Wireless Sensor Network. *Journal of Electrical and Computer Engineering*, 2014(January 2016):1–8, 2014.

[76] Shen Liu, James McGree, Zongyuan Ge, and Yang Xie. Classification methods. In *Computational and Statistical Methods for Analysing Big Data with Applications*, volume 65, pages 7–28. Elsevier, 2016.

[77] Saurabh Mukherjee and Neelam Sharma. Intrusion Detection using Naive Bayes Classifier with Feature Reduction. *Procedia Technology*, 4(March 2016):119–128, 2012.

[78] Xenia Naidenova and Lisa Tosheff. *Machine Learning Methods for Commonsense Reasoning Processes: Interactive Models InformatIon scIence reference*. 2010.

[79] Shubhangi S Gujar and B M Patil. I Ntrusion D Etection Using N Aïve B Ayes for R Eal. *International Journal of Advances in Engineering & Technology*, 7(2):568–574, 2014.

[80] V. Vijayakumar and V. Neelanarayanan. Preface. *Smart Innovation, Systems and Technologies*, 49:v–vii, 2016.

[81] Mayank Swarnkar and Neminath Hubballi. OCPAD: One class Naive Bayes classifier for payload based anomaly detection. *Expert Systems with Applications*, 64:330–339, 2016.

[82] Stephen Marsland. *M AC H I N E Edition*. 2015.

[83] Ankur A Patel. *Hands-On Unsupervised Learning Using Python_ How to Build Applied Machine Learning Solutions from Unlabeled Data-*. O'Reilly Media (2019), 2019.

[84] Junjie Wu. K-means Based Consensus Clustering. pages 155–175. 2012.

[85] Marco Capó, Aritz Pérez, and José Antonio Lozano. An efficient K-means algorithm for Massive Data. may 2016.

[86] T.M Kodinariya and P.R Makwana. Review on determining number of Cluster in K-Means Clustering. *International Journal of Advance Research in Computer Science and Management Studies*, 1(6):90–95, 2013.

[87] Jason Xu and Kenneth Lange. Power k -Means Clustering. 2019.

[88] Noe Elisa, Longzhi Yang, Yanpeng Qu, and Fei Chao. A Revised Dendritic Cell Algorithm Using K-Means Clustering. *Proceedings - 20th International Conference on High Performance Computing and Communications, 16th International Conference on Smart City and 4th International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2018*, pages 1547–1554, 2019.

[89] J. V. Anand Sukumar, I. Pranav, M. M. Neetish, and Jayasree Narayanan. Network Intrusion Detection Using Improved Genetic k-means Algorithm. *2018 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2018*, pages 2441–2446, 2018.

[90] Parisa Movahedi, Paavo Nevalainen, Markus Viljanen, and Tapio Pahikkala. Fast Regularized Least Squares and k-means Clustering Method for Intrusion Detection Systems. pages 264–269, 2015.

[91] Hsiang-chuan Liu, Bai-cheng Jeng, Jeng-ming Yih, and Yen-kuei Yu. <Isip09P422_2.Pdf>. 2, 2009.

[92] Adrian Stetco, Xiao Jun Zeng, and John Keane. Fuzzy C-means++: Fuzzy C-means with effective seeding initialization. *Expert Systems with Applications*, 42(21):7541–7548, 2015.

[93] Bruce McMillin. *Software Engineering*, volume 51. Springer Singapore, 2018.

[94] S Songma, W Chimphlee, K Maichalernnukul, and P Sanguansat. Implementation of Fuzzy c -Means and Outlier Detection for Intrusion Detection with KDD Cup 1999 Data Set. 2(2):44–48, 2012.

[95] Jingping Song, Zhiliang Zhu, Peter Scully, and Chris Price. Selecting features for anomaly intrusion detection: A novel method using fuzzy C means and decision tree classification. *Lecture Notes in Computer Science (including subseries Lecture Notes*

in Artificial Intelligence and Lecture Notes in Bioinformatics), 8300 LNCS:299–307, 2013.

[96] Rachna Kulhare and Divakar Singh. Intrusion Detection System based on Fuzzy C Means Clusteringand Probabilistic Neural Network. *International Journal of Computer Applications*, 74(2):30–33, 2013.

[97] Richa Sampat and Shilpa Sonawani. Network intrusion detection using dynamic fuzzy c means clustering. *Network*, 2(4):135–141, 2015.

[98] B.S. Harish and S.V.A. Kumar. Anomaly based Intrusion Detection using Modified Fuzzy Clustering. *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(6):54, 2017.

[99] Shawq Malik Mehibs and Soukaena Hassan Hashim. Proposed Network Intrusion Detection System Based on Fuzzy c Mean Algorithm in Cloud Computing Environment. *Journal of University of Babylon*, 26(2):27–35, 2017.

[100] Rana Aamir Raza Ashfaq, Xi Zhao Wang, Joshua Zhexue Huang, Haider Abbas, and Yu Lin He. Fuzziness based semi-supervised learning approach for intrusion detection system. *Information Sciences*, 378:484–497, 2017.

[101] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting Semi-Supervised Learning with Graph Embeddings. 48, 2016.

[102] Abdulla Amin Aburomman and Mamun Bin Ibne Reaz. A survey of intrusion detection systems based on ensemble and hybrid classifiers. *Computers and Security*, 65:135–152, 2017.

[103] Liang Shen, Qingsong Xu, Dongsheng Cao, Yizeng Liang, and Hongshuai Dai. The hybrid of semisupervised manifold learning and spectrum kernel for classification. *Journal of Chemometrics*, 32(2):1–11, 2018.

[104] Wathiq Laftah Al-Yaseen, Zulaiha Ali Othman, and Mohd Zakree Ahmad Nazri. Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. *Expert Systems with Applications*, 67:296–303, 2017.

[105] Chundong Wang, Xin Ye, Xiaonan He, and Yunkun Tian. *Security and Privacy in New Computing Environments*, volume 284. Springer International Publishing, 2019.

[106] John Vacca. *Computer and Information Security Handbook*. Number Mm. 2017.

[107] Donghao Zhou, Zheng Yan, Yulong Fu, and Zhen Yao. A survey on network data collection. *Journal of Network and Computer Applications*, 116(December 2017):9–23, 2018.

[108] Henry Hexmoor. *Computational Network Science*. 2016.

[109] Maoguo Gong, Qing Cai, Lijia Ma, Shanfeng Wang, and Yu Lei. *Computational Intelligence for Network Structure Analytics*. 2017.

[110] Uttam Kumar. A Survey on Intrusion Detection Systems for Cloud Computing Environment. 109(1):6–15, 2015.

[111] A Machine Learning. *Network Anomaly Detection: ML Perspective*. CRC Press, Boca Raton, Florida, 2014.

[112] Leonardo Serna-Guarín and Edilson Delgado-Trejos. Caracterización de canales de comunicación por tráfico y arquitectura de la red: una revisión
Characterization of communication channels in terms of traffic and network architecture: a review. *Iteckne*, 11(1):99–107, 2014.

[113] Pablo Vázquez Gil, Jorge Baeza, Pomares, and Francisco Herias, Candelas. *Redes y transmisión de datos*. Alicante, 2010.

[114] Leonardo Serna Guarín, Luis Javier Morantes Guzmán, and Edilson Delgado Trejos. *Transferencia óptima de datos para el monitoreo y control remoto de sistemas en tiempo real*. Instituto Tecnológico Metropolitano, Medellín, noviembre edition, nov 2015.

[115] Jeremy Faircloth. Networks. In *Enterprise Applications Administration*, volume 365, pages 27–79. Elsevier, 2014.

[116] Mohamed Abomhara and Geir M. K ien. Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks. *Journal of Cyber Security and Mobility*, 4(1):65–88, 2015.

[117] Chirag Modi, Dhiren Patel, Bhavesh Borisaniya, and Hiren Patel. Journal of Network and Computer Applications A survey of intrusion detection techniques in Cloud. 36:42–57, 2013.

[118] Mohamed Al-Hemairy, Saad Amin, and Zouheir Trabelsi. Towards more sophisticated ARP spoofing detection/prevention systems in LAN networks. *Proceedings of the 2009 International Conference on the Current Trends in Information Technology, CTIT 2009*, pages 225–230, 2009.

[119] Bharath Reddy S, D Malathi, and Shijoe Jose. An intrusion detection and prevention system in cloud computing_.pdf. *ARPN Journal of Engineering and Applied Sciences*, 12(12):3723–3729, 2017.

[120] Zuherman Rustam and Aini Suri Talita. Fuzzy kernel c-means algorithm for intrusion detection systems. *Journal of Theoretical and Applied Information Technology*, 81(1):161–165, 1 2015.

[121] James Cannady. Artificial Neural Networks for Misuse Detection. *Proceedings of the 21st National Information Systems Security Conference*, pages 443–446, 1998.

[122] Akash Garg and Prachi Maheshwari. Performance analysis of Snort-based Intrusion Detection System, 2016.

[123] D. A Orrego Metaute. *Análisis de relevancia usando fuzzy rough sets orientado a la reducción de dimensiones de espacios de representación característica en aplicaciones biomédicas*. PhD thesis, Instito Tecnológico Metrpolitano, 2012.

[124] Afsaneh Mahanipour, Hossein Nezamabadi-Pour, and Bahareh Nikpour. Using fuzzy-rough set feature selection for feature construction based on genetic programming. *3rd Conference on Swarm Intelligence and Evolutionary Computation, CSIEC 2018*, pages 1–6, 2018.

[125] Richard Jensen and Qiang Shen. *Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches*. 2008.

[126] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning—I. *Information Sciences*, 8(3):199–249, jan 1975.

[127] Neil Mac Parthaláin and Richard Jensen. Measures for unsupervised fuzzy-rough feature selection. *ISDA 2009 - 9th International Conference on Intelligent Systems Design and Applications*, 7:560–565, 2009.

[128] Igor Kononenko. Estimating attributes: Analysis and extensions of RELIEF. pages 171–182, 1994.

[129] Kenji Kira and Larry A. Rendell. *A Practical Approach to Feature Selection*. Morgan Kaufmann Publishers, Inc., 2014.

[130] Raul Jose Palma-Mendoza, Daniel Rodriguez, and Luis De-Marcos. Distributed ReliefF-based feature selection in Spark. *Knowledge and Information Systems*, 57(1), 2018.

[131] Zheng Alan Zhao and Huan Liu. *Spectral Feature Selection for Data Mining*. Chapman and Hall/CRC, dec 2011.

[132] M Robnik and I Konenko. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning*, 53(1–2):23–69, 2003.

[133] Ronald N. Forthofer, Eun Sul Lee, and Mike Hernandez. 9 - nonparametric tests. In Ronald N. Forthofer, Eun Sul Lee, and Mike Hernandez, editors, *Biostatistics (Second Edition)*, pages 249 – 268. Academic Press, San Diego, second edition edition, 2007.

[134] VLADIMIR CORTES, CORINNA, VAPNIK. In silico log P prediction for a large data set with support vector machines, radial basis neural networks and multiple linear regression. *Chemical biology & drug design*, 20(2):273–297, aug 1995.

[135] Jayadeva, Reshma Khemchandani, and Suresh Chandra. *Twin Support Vector Machines*, volume 659 of *Studies in Computational Intelligence*. Springer International Publishing, Cham, apr 2017.

[136] Catalin Stoean and Ruxandra Stoean. *Support Vector Machines and Evolutionary Algorithms for Classification*, volume 69 of *Intelligent Systems Reference Library*. Springer International Publishing, Cham, 2014.

[137] Solangel Rodríguez-vázquez and Andy Vidal Martínez-borges. Clasificación de células cervicales con Máquinas de Soporte Vectorial empleando rasgos del núcleo Cervical cell classification with Support Vector Machines using nucleus ' features. 9(2):115–127, 2015.

[138] Raoof Gholami and Nikoo Fakhari. *Support Vector Machine: Principles, Parameters, and Applications*. Elsevier Inc., 1 edition, 2017.

[139] Alvaro Flores, Sebastián Maldonado, and Richard Weber. Selección de Atributos y Support Vector Machines Adaptado al problema de Fuga de Clientes. *Revista Ingenieria de Sistemas*, pages 87–110, 2015.

[140] E Boser, N Vapnik, Isabelle M Guyon, and T Bell Laboratories. p144-boser.pdf (application/pdf Object). pages 144–152, 1992.

[141] Jakub Nalepa and Michal Kawulok. Selecting training sets for support vector machines: a review. *Artificial Intelligence Review*, pages 1–44, 2018.

[142] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, volume . Cambridge University Press, Cambridge, 2014.

[143] Sergio Herrero-Lopez. Multiclass Support Vector Machine. In *GPU Computing Gems Emerald Edition*, volume 9783319023, pages 293–311. Elsevier, 2011.

[144] Christoph H. Lampert. Kernel Methods in Computer Vision. *Foundations and Trends® in Computer Graphics and Vision*, 4(3):193–285, 2009.

[145] Bo Liu, Yanshan Xiao, and Longbing Cao. SVM-based multi-state-mapping approach for multi-class classification. *Knowledge-Based Systems*, 129:79–96, 2017.

[146] David Hutchison. *Future Data and Security Engineering*, volume 10018 of *Lecture Notes in Computer Science*. Springer International Publishing, Cham, 2016.

[147] J. C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973.

[148] James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*, volume . Springer US, Boston, MA, 1981.

[149] S Kalyani and K Swarup. Supervised fuzzy C-means clustering technique for security assessment and classification in power systems. *International Journal of Engineering, Science and Technology*, 2(3):175–185, 2011.

[150] Waleed Alomoush, Ayat Alrosan, Norita Norwawi, Yazan Alomari, Dheeb Albashish, Ammar Almomani, and Mohammed Alqahtani. A survey: Challenges of image segmentation based fuzzy c-means clustering algorithm. *Journal of Theoretical and Applied Information Technology*, 96(16):5153–5170, 2018.

[151] Moslem Fatehi and Hooshang H. Asadi. Application of semi-supervised fuzzy c-means method in clustering multivariate geochemical data, a case study from the Dalli Cu-Au porphyry deposit in central Iran. *Ore Geology Reviews*, 81:245–255, 2017.

[152] Daphne Teck Ching Lai and Jonathan M. Garibaldi. Improving semi-supervised fuzzy c-means classification of Breast Cancer data using feature selection. *IEEE International Conference on Fuzzy Systems*, 2013.

[153] Yasunori Endo, Yukihiro Hamasuna, Makito Yamashiro, and Sadaaki Miyamoto. On semi-supervised fuzzy c-means clustering. *IEEE International Conference on Fuzzy Systems*, 1:1119–1124, 2009.

[154] Chunfang Li, Lianzhong Liu, and Wenli Jiang. Objective function of semi-supervised FUZZY C-Means clustering algorithm. *IEEE International Conference on Industrial Informatics (INDIN)*, (1):737–742, 2008.

[155] Chiheb Chebbi. *Mastering machine learning for penetration testing : develop an extensive skill set to break self-learning systems using Python*. 2018.

[156] Amani Yahyaoui and Imene Yahyaoui. *Machine Learning Techniques for Data Classification*, volume 2. Elsevier Inc., 2018.

[157] Xin-She Yang. Data mining techniques. In *Introduction to Algorithms for Data Mining and Machine Learning*, volume 1, pages 109–128. Elsevier, 2019.

[158] Zichan Ruan, Yuantian Miao, Lei Pan, Nicholas Patterson, and Jun Zhang. Visualization of big data security: a case study on the KDD99 cup data set. *Digital Communications and Networks*, 3(4):250–259, 2017.

[159] Laheeb M Ibrahim, Dujan T Basheer, and Mahmod S Mahmod. (Kdd99 , Nsl-Kdd) Based on Self Organization Map (Som) Artificial Neural Network. *Journal of Engineering Science and Technology*, 8(1):107–119, 2013.

[160] Danijela Protić. Review of KDD Cup '99, NSL-KDD and Kyoto 2006+ datasets, 2018.

[161] Vivek Nandan Tiwari, Satyendra Rathore, and Kailash Patidar. International Journal of Current Trends in Engineering & Technology Enhanced Method for Intrusion Detection over KDD Cup 99 Dataset. *Mar-Apr*, 02:2–2, 2016.

[162] Abhishek Divekar, Meet Parekh, Vaibhav Savla, Rudra Mishra, and Mahesh Shirole. Benchmarking datasets for Anomaly-based Network Intrusion Detection: KDD CUP 99 alternatives. *Proceedings on 2018 IEEE 3rd International Conference on Computing, Communication and Security, ICCCS 2018*, pages 1–8, 2018.

[163] Ahmed Ahmim, Makhlouf Derdour, and Mohamed Amine Ferrag. An intrusion detection system based on combining probability predictions of a tree of classifiers. *International Journal of Communication Systems*, 31(9):1–17, 2018.

[164] Kulkarni Parag A. *Pattern based network security.* PhD thesis, 2012.

[165] Aslıhan AKYOL, Mehmet HACIBEYOĞLU, and Bekir KARLIK. Design of Multilevel Hybrid Classifier with Variant Feature Sets for Intrusion Detection System. *IEICE Transactions on Information and Systems*, E99.D(7):1810–1821, 2016.