Analisis de Datos para Ciberseguridad: Trabajo práctico 2

Ph. D. Saúl Calderón Ramírez
Instituto Tecnológico de Costa Rica,
Escuela de Ingeniería en Computación, Programa de Ciencias de Datos,
PAttern Recongition and MAchine Learning Group (PARMA-Group)

27 de agosto de 2025

Fecha de entrega: Domingo 14 de Setiembre.
Entrega: Un archivo .zip con el código fuente LaTeX o Lyx, el pdf, y un notebook Jupyter, debidamente documentado. A través del TEC-digital.
Modo de trabajo: Grupos de 3 personas.

Resumen

En el presente trabajo práctico se introducir los arboles de decision por medio de su implementación para resolver un problema práctico en ciberseguridad.

En el presente trabajo practico se utilizara el *dataset KDD99* `https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`. El conjunto de datos KDD 1999 es uno de los más utilizados para entrenar y evaluar sistemas de detección de intrusos (IDS). Está basado en simulaciones de tráfico de red en un entorno militar y contiene conexiones etiquetadas como normales o como uno de varios tipos de ataques.

1. Categorías de características (features): Las 41 características del dataset se dividen en cuatro grupos principales:
   a) Características básicas (Basic features): Información general de la conexión como duración, protocolo, servicio, estado de la conexión. Ejemplos: duration, protocol_type, service, flag.
   b) Características de contenido (Content features): Analizan el contenido de la conexión para detectar intentos de acceso no autorizado. Ejemplos: num_failed_logins, logged_in, root_shell, num_compromised.
   c) Características de tráfico basado en tiempo (Time-based traffic features): Estadísticas de conexiones en una ventana de tiempo (por ejemplo, en los últimos 2 segundos). Ejemplos: count, srv_count, serror_rate, srv_serror_rate.

d) Características de tráfico basado en host (Host-based traffic features): Estadísticas de conexiones hacia el mismo host en una ventana más amplia.- Ejemplos: dst_host_count, dst_host_srv_count, dst_host_same_srv_rate.

2. Las conexiones se etiquetan como normales o como uno de los siguientes tipos de ataque:

   a) DoS (Denial of Service): Saturar el sistema para que no responda a usuarios legítimos. Ej: smurf, neptune.

   b) Probe: Recolección de información sobre la red. Ej: satan, portsweep.

   c) R2L (Remote to Local): Acceso no autorizado desde una máquina remota. Ej: guess_passwd, warezclient.

   d) U2R (User to Root): Escalada de privilegios desde una cuenta local. Ej: buffer_overflow, rootkit.

Para el presente trabajo practico se usaran todas las caracteristicas para estimar las 21 categorias de ataque o normalidad del mensaje. Como preprocesamiento, se incluye la conversion de atributos nominales a numericos, y la eliminacion de registros faltantes y redudantes.

1. (30 puntos) Implementacion de un arbol de decision y random forests para clasificar todos los tipos de ataques (scikit learn)

1. Genere una funcion *split_dataset* la cual divida los datos en entrenamiento (70 %), validacion (15 %) y prueba (15 %).

2. **(15 puntos)** Entrene un arbol de decision (de *scikitlearn*) para clasificar los ataques en todas las categorias originales del dataset:

   a) Optimice la profundidad maxima, cantidad minima de observaciones por particion, y el criterio de pureza usando *optuna o weights and biases.* Documente los rangos de cada hiper-parametro y justifique la decision por cada uno. Muestre los graficos de ese proceso de optimizacion y seleccione las 3 mejores arquitecturas.

   b) Compare las tres mejores arquitecturas, para al menos 10 corridas diferentes (particiones), el F1-score promedio para todas las clases y la tasa de falsos positivos promedio para todas las clases, presente medias y desviaciones estandar usando la particion de prueba. Comente los resultados.

3. **(15 puntos)** Usando tambien la libreria *scikit learn* entrene un *random forest*.

a) Optimice con optuna o weights and biases la cantidad de arboles del *random forest*. Defina el rango de numero de arboles de decision de forma justificada.

b) Compare los 2 mejores *random forests* seleccionados por la herramienta, haciendo 10 corridas diferentes (particiones), el F1-score promedio para todas las clases y la tasa de falsos positivos promedio para todas las clases, presente medias y desviaciones estandar usando la particion de prueba. Comente los resultados.

2. (30 puntos) Implementacion de una red neuronal para la clasificacion de todos los tipos de ataques (pytorch)

1. **(10 puntos)** Implemente la clase *FCN* la cual implemente una red de 4 capas en pytorch, y defina la mejor arquitectura posible según las restricciones del problema, de forma justificada. Justifique que funciones de activación son las más pertinentes para el problema.

   a) Use en tal arquitectura la cantidad de neuronas/capas que usted desee, y como funciones de activacion, en las capas intermedias funciones de activacion ReLU, en la capa de salida la funcion de activacion *softmax* .

   b) Implemente la funcion *train_fcn* la cual tome el modelo previamente creado y el conjunto de datos de entrenamiento, y entrene tal modelo:

      1) Usando *optuna* o *weights and biases*, calibre el coeficiente de aprendizaje. Muestre las graficas de calibracion e indique el coeficiente de aprendizaje.
      2) Pruebe al menos 3 variantes posibles de arquitectura (justifique la decision de cada aspecto usando articulos de referencia de ser posible), y muestre la grafica del error de entrenamiento y validacion, para justificar la seleccion del modelo propuesto.

2. **(10 puntos)** Implemente la funcion *evaluate_fcn* la cual para la particion de datos de prueba compute la tasa de aciertos, el f1-score promediado, y ademas muestre la matriz de confusion.

   a) Evalue los 3 modelos propuestos anteriormente, realizando al menos 10 particiones aleatorias de entrenamiento y prueba, y presenta una tabla con todos los resultados, media y desviacion estandar del F1-score y la tasa de falsos negativos promedio, de cada modelo propuesto.

3. **(5 puntos)** En la funcion *train_fcn* implemente el pesado por cantidad de observaciones por clase comentado en clase, donde se le de un peso mayor a las observaciones de las clases sub-representadas.

    a) Compare la mejor arquitectura seleccionada en el proceso anterior usando y no usando el pesado de la funcion de perdida realizando al menos 10 particiones aleatorias de entrenamiento y prueba, y presenta una tabla con todos los resultados, media y desviacion estandar de cada modelo propuesto.

4. **(5 puntos)** Compare todos los metodos implementados usando como referencia los resultados. Argumente las ventajas y desventajas de cada uno para justificar la decision del mejor modelo.

3. (15 puntos extra) Implementacion de una TabNet para la clasificacion de todos los tipos de ataques (pytorch)

1. **(5 puntos)**Investigue y explique como funciona la arquitectura TabNet para clasificar datos estructuradores, usando como referencia inicial el articulo *Tabnet: Attentive interpretable tabular learning*. Base su investigacion a articulos recientes publicados en conferencias y revistas, y use el diagrama de la arquitectura para explicar su funcionamiento. Argumente sus ventajas para la clasificacion de datos estructurados.

2. **(15 puntos)** Seleccione de forma justificada los hiper-parametros mas relevantes para TabNet, y calibrelos con *optuna* o *weights and biases*. Seleccione las 2 mejores configuraciones usando como base los graficos generados en la calibracion. Guarde los parametros de los 2 mejores modelos para facilitar la evaluacion de modelos que tardan mucho en su entrenamiento.

    a) Evalue los 2 modelos propuestos anteriormente, cada uno con y sin el pesado de las observaciones segun la prevalencia de su clase, implementado en la seccion anterior (4 modelos en total), realizando al menos 10 particiones aleatorias de entrenamiento y prueba, y presenta una tabla con todos los resultados, media y desviacion estandar del F1-score y la tasa de falsos negativos promedio, de cada modelo propuesto.

        1) Comente los resultados, y exponga las ventajas y desventajas de este modelo con los anteriormente probados.