

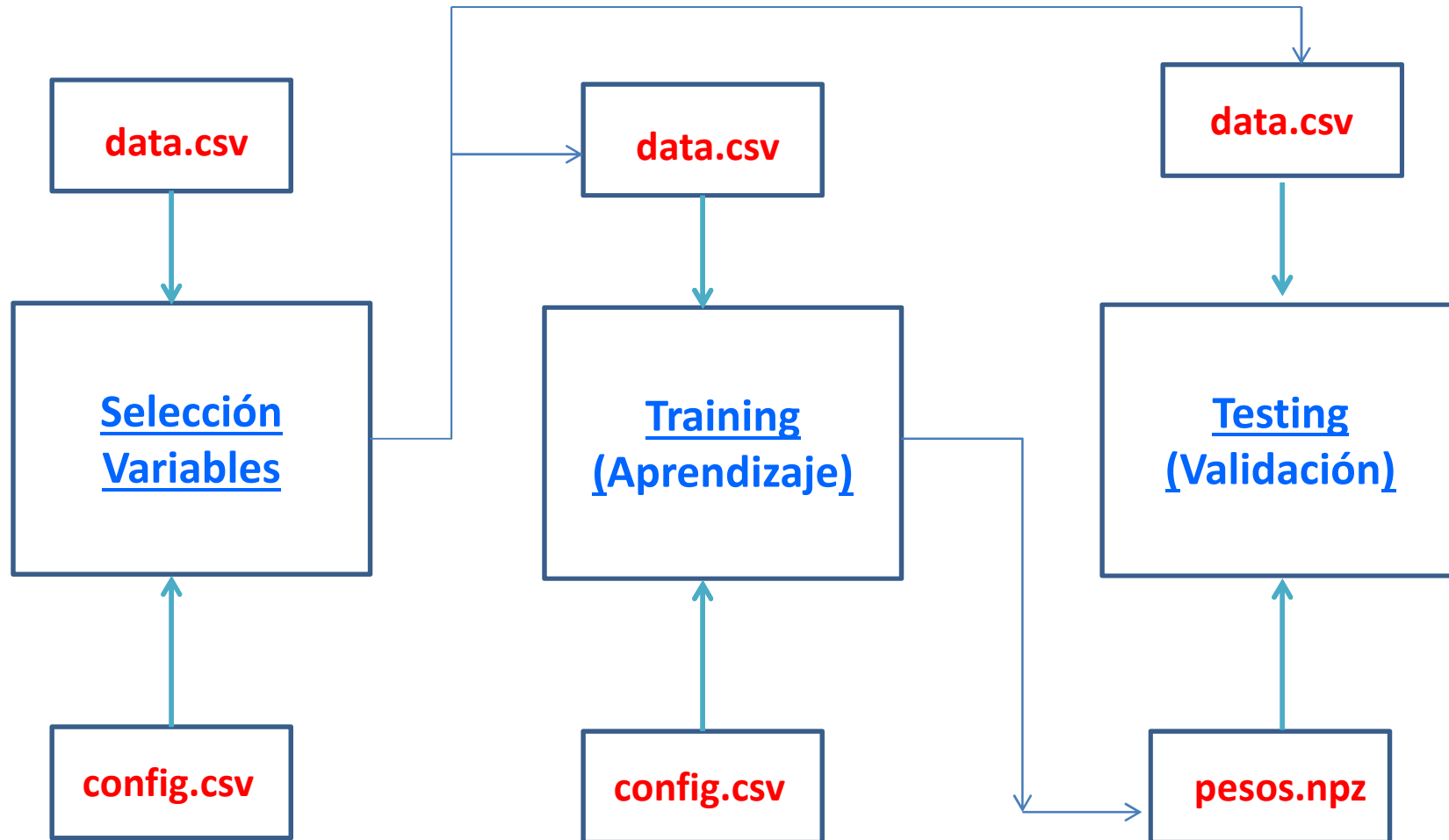
# **EVALUACIÓN #3**

Prof. Nibaldo Rodríguez A.

# **OBJETIVOS**

- **General:**
  - Implementar y evaluar un sistema de detección de intrusos usando técnicas machine learning.
- **Específicos:**
  - Seleccionar variables usando ganancia de información, información mutua y valores propios de matrices.
  - Implementar una red neuronal artificial (ANN) usando aprendizaje híbrido (PSO+BP) para clasificar tres tipo de tráfico (normal, dos, probe).
  - Evaluar el rendimiento de la ANN usando métricas de exactitud y F-scores para cada clase.

# Etapas del Modelo:



## Selección Variables:

- **EQUIPO #A:**
  - **Grupos: 1 al 8**
    - Ganancia de la Información
    - SVD
- **EQUIPO #B:**
  - **Grupos: 9 al 15**
    - Información Mutua
    - SVD

## Selección Variables: Data

- Datos de Training: **KDDTrain.txt**
  - Muestras: 25192, Variables: 43
  - Variable Número 42: tipo de tráfico
    - Normal, Clase 1
    - DOS, Clase 2
    - Probe, Clase 3
- Datos de Testing: **KDDTest.txt**
  - Muestras: 22544, Variables: 43
  - Clases: {1,2,3}.

## Selección Variables: Data

- **Clase #1: Valores**
  - 'normal'
- **Clase #2: Valores**
  - 'neptune', 'teardrop', 'smurf', 'pod', 'back',
  - 'land', 'apache2', 'processtable', 'mailbomb',
  - 'udpstorm'
- **Clase #3: Valores**
  - 'ipsweep', 'portsweep', 'nmap',
  - 'satan', 'saint', 'mscan'

## Selección Variables: *sv.py*

- **Convertir Variables no-numéricas a numéricas:**
- **Variables categóricas:**
  - Var #2: protocolo
  - Var #3: servicio
  - Var #4: flag
- **Variables continuas**
  - Vars = { 1, 5:41 }
- **Variable Etiqueta**
  - Var #42= { 1 ó 2 ó 3 }

## Selección Variables: *sv.py*

- Normalizar cada variable = {1:41}

$$x = \frac{(x - x_{\min})}{(x_{\max} - x_{\min})} \times (b - a) + a, \quad a = 0.01 \quad b = 0.99$$



## Selección Variables: sv.py

- Equipo #A:
  - Calcular la data de relevancia usando el método de ganancia de información .
- Equipo #B:
  - Calcular la data de relevancia usando el método de Información Mutua.
- Crear archivo con los índices relevantes :
  - **index.csv** : M-filas por 1-columna
- Calcular la reducción de redundancia usando el método de descomposición de valores singulares.
  - Crear archivo con la matriz de filtro (matriz  $V$ ) :
    - **filter\_v.csv**.

## Selección Variables: sv.py

- Filtrar la data usando archivo **filter\_v.csv**
- Crear archivo de datos con data filtrada:
  - **dtrn.csv**: data de entrada para la red neuronal
  - **etrn.csv**: data de etiquetas para la red.

## Selección Variables: sv.py

- Cargar Datos de Testing: **KDDTest.txt**
- Convertir variables no-numéricas a numéricas
- Normalizar cada variables {1:41} usando método previo.
- Filtrar la data normaliza usando los archivos **index.csv**, **filter\_v.csv**.
  - Normalizar la data filtrada.
- Crear archivo con datos normalizados:
  - **dtst.csv**

## Selección Variables: sv.py

- Crear Etiquetas numéricas para Variable #42:
  - Caso #1 : Normal
  - Caso #2 : DOS
  - Caso #3 : Probe
- Crear archivo de Etiquetas de testing:
  - etst.csv

## Configuración : Selección de Variables

- **cnf\_sv.csv**
- Número de Muestras de Train : 1000
- Número de Muestras de Test : 5000
- Valor de Relevancia (0,1) : 0.6
- Número de Vectores Singulares : 10
- Clase Normal (s/n) : 1
- Clase DOS (s/n) : 0
- Clase Probe (s/n) : 1.

## Entrenamiento ANN : trn.py

- Re-ordenar aleatoriamente la posición de la Data:
  - **dtrn.csv**            **etrn.csv**
- Crear Etiquetas Binarias: **etrn.csv**
  - **Caso #1: Normal, DOS**
  - **Caso #2: Normal, Probe**
  - **Caso #3: Normal, DOS, Proble.**

## Entrenamiento ANN : trn.py

Entrenar la ANN+PSO

+

Entrenar la ANN+BP

## Entrenamiento ANN : **tst.py**

- Re-ordenar aleatoriamente la posición de la Data:
  - **dtst.csv**            **etst.csv**
- Crear Etiquetas Binarias: **etst.csv**
  - **Caso #1: Normal, DOS**
  - **Caso #2: Normal, Probe**
  - **Caso #3: Normal, DOS, Proble.**



## Configuración : ANN-PSO

- **cnf\_ann\_pso.csv**
- Línea 1 : Número Función Oculta : 5
- Línea 2 : Número Función Salida : 8
- Línea 3 : Número Nodos Ocultos : 20
- Línea 4 : Número Iteraciones : 1000
- Línea 5 : Número de Partículas : 20
- ...
- ...

## Configuración : ANN-BP

- **cnf\_ann\_bp.csv**
- Línea 1 : Número Iteraciones : 2000
- Línea 2 : Tasa de Aprendizaje : 0.1
- ...
- ...

**RESULTADOS :**

**Train/ Test**

## Train: trn.py

- Crear archivos:
  - **costo\_pso.csv**
    - N-Filas por 1-Columnas
    - Cada fila representa el valor del Error Cuadrático Medio (MSE) durante la etapa de aprendizaje.
- Crear archivo binario de pesos:
  - **pesos\_pso.npz**
    - **w1 ,w2**

## Train: trn.py

- Crear archivos:
  - **costo\_bp.csv**
    - N-Filas por 1-Columnas
    - Cada fila representa el valor del Error Cuadrático Medio (MSE) durante la etapa de aprendizaje.
- Crear archivo binario de pesos:
  - **pesos\_bp.npz**
    - **w1 ,w2**

## Test: tst.py

- Matriz de Confusión:
  - **cmatrix.csv**
    - Cada elemento representa el valor predicho versus el valor real.
    - Filas: Valor predicho
    - Columnas: Valor Real.
- Fscores:
  - **fscores.csv**
    - 4-Filas por 1-Columna
    - Cada Fila:
      - » F-scores de la Clase 1
      - » F-scores de la Clase 2
      - » F-scores de la Clase 3
      - » F-scores promedio.

**Lenguaje de Programación:**

**Python- Numpy ( 3.7)**

## **Fechas Entrega:**

- Lunes 07/11/2022, Hora: 09:00 am:
  - Selección de variables
- Lunes 21/11/2022, Hora: 09:00 am
  - Clasificación para tres tipo de tráfico
- **Programas fuentes:**
  - sv.py
  - trn.py
  - tst.py
  - myutility.py
- **Archivos de Configuración:**
  - cnf\_sv.csv
  - cnf\_ann\_pso.csv
  - cnf\_ann\_bp.csv .



## **Fechas Entrega:**

- **Archivos de Resultados:**

- index.csv
- filter\_v.csv
- costos\_pso.csv
- costos\_bp.csv
- cmatrix.csv
- fscore.csv

- **Nombre archivo Entregable:**

- grupoNum.zip

## **OBSERVACIÓN:**

- Si un Grupo no Cumple con los requerimientos funcionales y no-funcionales, entonces será evaluado con nota entre 1 y 3.
- Si un Grupo es hallado que ha realizado copia de los programas fuentes, entonces será evaluado con la nota mínima.