

Diseño de un modelo neuronal para la detección y la clasificación de intrusiones en redes informáticas

Hugo López Álvarez
Tutor: Diego García Álvarez

20 de junio de 2025



Universidad de Valladolid

- 1. Introducción
- 2. Planificación y costes
- 3. Metodología
- 4. Entendimiento del problema
- 5. Entendimiento de los datos
- 6. Modelado
- 7. Evaluación
- 8. Despliegue
- 9. Conclusiones

Índice

1. Introducción

2. Planificación y costes

3. Metodología

4. Entendimiento del problema

5. Entendimiento de los datos

6. Modelado

7. Evaluación

8. Despliegue

9. Conclusiones

Contexto

- ▶ Intrusiones en sistemas informáticos.
- ▶ Evolución de los ataques a las redes informáticas como consecuencia del uso de IA.
- ▶ Defensa antes posibles intrusiones.

Objetivos del proyecto

- ▶ Diseñar e implementar un modelo capaz de detectar intrusiones en redes informáticas y proporcionar una clasificación previa de la intrusión.
- ▶ Los modelos de detección han de ser modelos neuronales.
- ▶ Evaluar y comparar los modelos generados con un dataset real y complejo .

Objetivos del académicos

- ▶ Comprender el funcionamiento de los modelos neuronales através de PyTorch y las métricas de evaluación.
- ▶ Aprender las características de varios de los tipo de modelos neuronales que existen.
- ▶ Descubrir el potencial de las redes neuronales para optimizar y mejorar las tecnologías de la información, incluyendo la ciberseguridad de los sistemas.

Índice

1. Introducción

2. Planificación y costes

3. Metodología

4. Entendimiento del problema

5. Entendimiento de los datos

6. Modelado

7. Evaluación

8. Despliegue

9. Conclusiones

Índice

1. Introducción

2. Planificación y costes

3. Metodología

4. Entendimiento del problema

5. Entendimiento de los datos

6. Modelado

7. Evaluación

8. Despliegue

9. Conclusiones

CRISP-DM

- ▶ Diseñada para guiar proyectos de minería de datos y aprendizaje automático.
- ▶ Su estructura cíclica y flexible, la hace aplicable en diversos dominios, desde marketing hasta ciberseguridad.

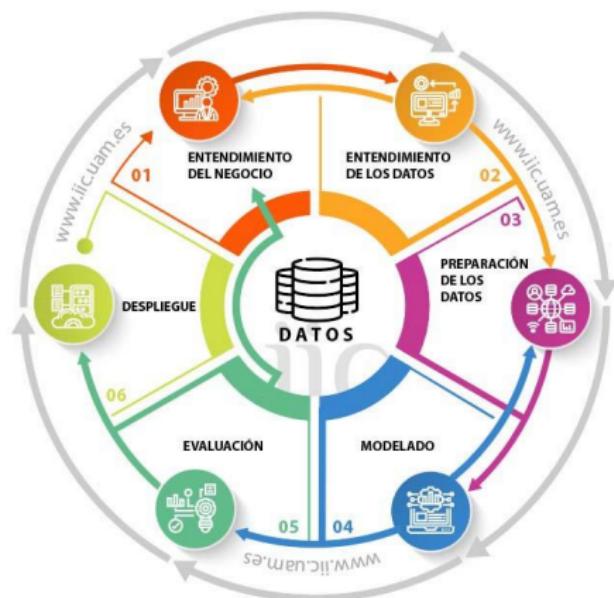


Figura: Esquema del ciclo CRISP-DM estándar [?].

Fases de CRISP-DM

1. Entendimiento del problema.
2. Entendimiento de los datos.
3. Preparación de los datos.
4. Modelado.
5. Evaluación.
6. Despliegue.

Índice

1. Introducción

6. Modelado

2. Planificación y costes

7. Evaluación

3. Metodología

8. Despliegue

4. Entendimiento del problema

9. Conclusiones

5. Entendimiento de los datos

Conceptos básicos

- ▶ ¿Qué es un ataque a un sistema informático?
- ▶ Tipos de ataque más comunes.
- ▶ TCP/IP.

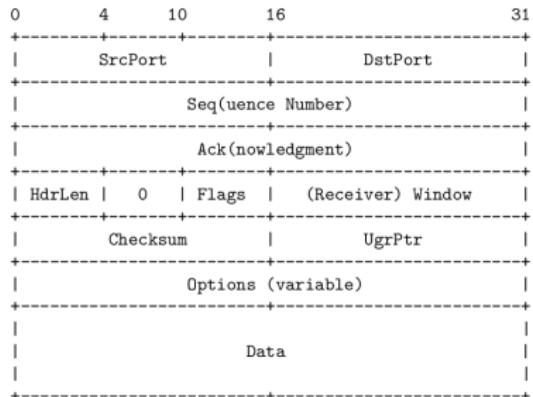


Figura: Esquema segmento TCP [?].

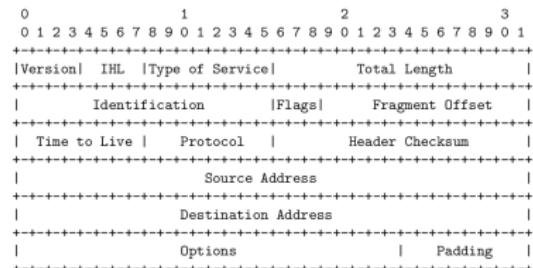


Figura: Formato de la cabecera IPv4 [?].

Prueba

- ▶ Importancia de protegerse frente a un ataque informático.
- ▶ Importancia de detectar rápidamente frente a un ataque informático.

Soluciones actuales

- ▶ Cortafuegos de próxima generación (NGFW).
- ▶ Sistema de detección de intrusiones (IDS).
- ▶ Sistema de prevención de intrusiones (IPS).

Índice

1. Introducción

2. Planificación y costes

3. Metodología

4. Entendimiento del problema

5. Entendimiento de los datos

Características de los
datos

Preparación de los datos

6. Modelado

7. Evaluación

8. Despliegue

9. Conclusiones

Origen de los datos

El dataset utilizado en este trabajo es NF-UNSW-NB15-v3, desarrollado como parte de un análisis realizado en la Universidad de Queensland, Australia. Contiene 53 atributos que describen características del tráfico de red y que permiten clasificar las muestras de tráfico en nueve clases de ataques o como conexiones benignas. Algunos de los datos que recoge son:

- ▶ La duración de la conexión.
- ▶ Los bytes enviados y los recibidos.
- ▶ Versiones de los protocolos utilizados.

Tipos de ataques registrados en el conjunto de datos

Clase	Cantidad
Benigno	2 237 731
<i>Fuzzers</i>	33 816
<i>Analysis</i>	2 381
<i>Backdoor</i>	1 226
DoS	5 980
<i>Exploits</i>	42 748
<i>Generic</i>	19 651
<i>Reconnaissance</i>	17 074
<i>Shellcode</i>	4 659
<i>Worms</i>	158

Cuadro: Clasificación de amenazas de seguridad del dataset
NF-UNSW-NB15-v3.

Atributos

Los atributos utilizados para desarrollar los modelos de este trabajo son los 53 atributos del conjunto de datos original, a excepción de los eliminados, que son:

- ▶ IPV4_SRC_ADDR
- ▶ IPV4_DST_ADDR
- ▶ FLOW_START_MILLISECONDS
- ▶ FLOW_END_MILLISECONDS

Etiquetas

1. Label: Indica si se trata de una conexión benigna o maligna.
2. Attack: Indica el tipo de ataque al que corresponde esa conexión.

Índice

1. Introducción

2. Planificación y costes

3. Metodología

4. Entendimiento del problema

5. Entendimiento de los datos

6. Modelado

Origen de los modelos neuronales

Clasificación con redes neuronales

Arquitecturas desarrolladas

Implementación de los modelos

Selección de las configuraciones
de los modelos

7. Evaluación

8. Despliegue

9. Conclusiones

Origen de los modelos neuronales

- ▶ Ramón y Cajal
- ▶ McCulloch/Pitts
- ▶ Rosenblatt



Figura: Ramón y Cajal.

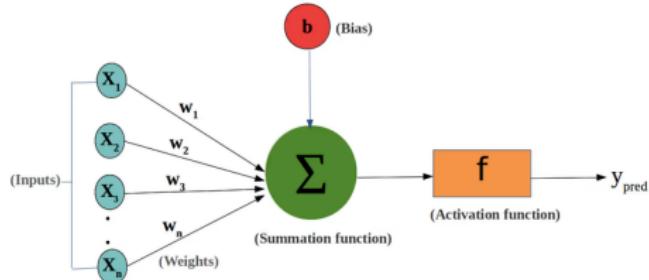


Figura: Esquema del funcionamiento de una neurona artificial [?].

Parámetros e hiperparámetros

Los parámetros se ajustan através del algoritmo de optimización utilizando los cálculos de la función de pérdida durante la fase de entrenamiento de los modelos. Los parámetros de una red neuronal son:

- ▶ Pesos
- ▶ Bias o sesgos

Los hiperparámetros son valores que se configuran antes de la fase de entrenamiento, que controlan el comportamiento del proceso de entrenamiento. Algunos de los hiperparámetros más importantes son:

- ▶ Tasa de aprendizaje (*Learning rate*)
- ▶ Épocas (*Epochs*)
- ▶ Tamaño de lote (*Batch size*)

Matrices de confusión para el MCB

- ▶ Verdaderos positivos (VP)
- ▶ Falsos positivos (FP)
- ▶ Falsos negativos (FN)
- ▶ Verdaderos negativos (VN)

	Predictión Positiva	Predictión Negativa
Real Positivo	VP	FN
Real Negativo	FP	VN

Cuadro: Matriz de confusión para modelos de clasificación binaria.

Matriz de confusión para el MCM

	Predicción Clase 1	Predicción Clase 2	Predicción Clase 3	Predicción Clase 4	Predicción Clase 5	Predicción Clase 6	Predicción Clase 7	Predicción Clase 8	Predicción Clase 9
Real Clase 1	VP ₁	FP ₁₂	FP ₁₃	FP ₁₄	FP ₁₅	FP ₁₆	FP ₁₇	FP ₁₈	FP ₁₉
Real Clase 2	FP ₂₁	VP ₂	FP ₂₃	FP ₂₄	FP ₂₅	FP ₂₆	FP ₂₇	FP ₂₈	FP ₂₉
Real Clase 3	FP ₃₁	FP ₃₂	VP ₃	FP ₃₄	FP ₃₅	FP ₃₆	FP ₃₇	FP ₃₈	FP ₃₉
Real Clase 4	FP ₄₁	FP ₄₂	FP ₄₃	VP ₄	FP ₄₅	FP ₄₆	FP ₄₇	FP ₄₈	FP ₄₉
Real Clase 5	FP ₅₁	FP ₅₂	FP ₅₃	FP ₅₄	VP ₅	FP ₅₆	FP ₅₇	FP ₅₈	FP ₅₉
Real Clase 6	FP ₆₁	FP ₆₂	FP ₆₃	FP ₆₄	FP ₆₅	VP ₆	FP ₆₇	FP ₆₈	FP ₆₉
Real Clase 7	FP ₇₁	FP ₇₂	FP ₇₃	FP ₇₄	FP ₇₅	FP ₇₆	VP ₇	FP ₇₈	FP ₇₉
Real Clase 8	FP ₈₁	FP ₈₂	FP ₈₃	FP ₈₄	FP ₈₅	FP ₈₆	FP ₈₇	VP ₈	FP ₈₉
Real Clase 9	FP ₉₁	FP ₉₂	FP ₉₃	FP ₉₄	FP ₉₅	FP ₉₆	FP ₉₇	FP ₉₈	VP ₉

Cuadro: Matriz de confusión para modelos de clasificación multiclas (9 clases).

Arquitecturas desarrolladas para el MCB

- ▶ **MCB25:** La mitad del número de atributos o entradas que recibe el modelo
- ▶ **MCB49:** El mismo número que atributos o entradas que tiene el modelo.
- ▶ **MCB98:** El doble del número de atributos o entradas.

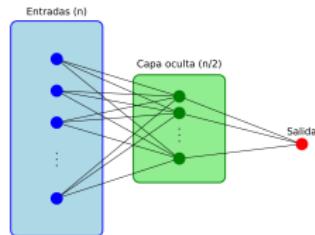


Figura: MCB25.

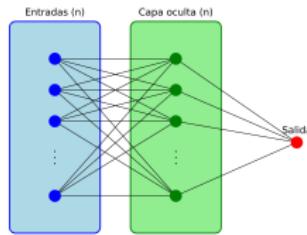


Figura: MCB49.

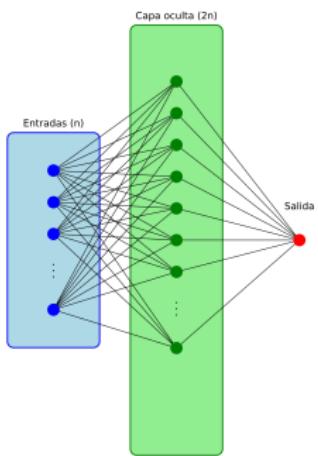


Figura: MCB98.

Arquitecturas desarrolladas para el MCM

- ▶ **MCM25:** La mitad del número de atributos o entradas que recibe el modelo
- ▶ **MCM49:** El mismo número que atributos o entradas que tiene el modelo.
- ▶ **MCM98:** El doble del número de atributos o entradas.

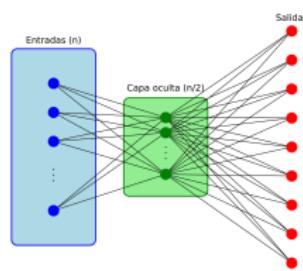


Figura: MCM25.

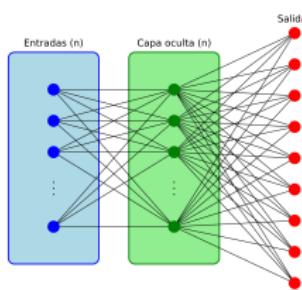


Figura: MCM49.

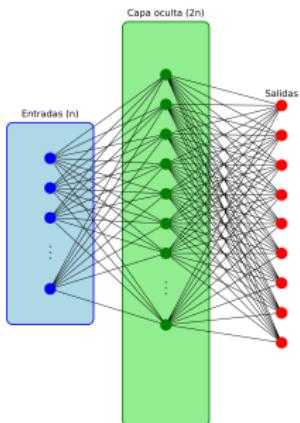
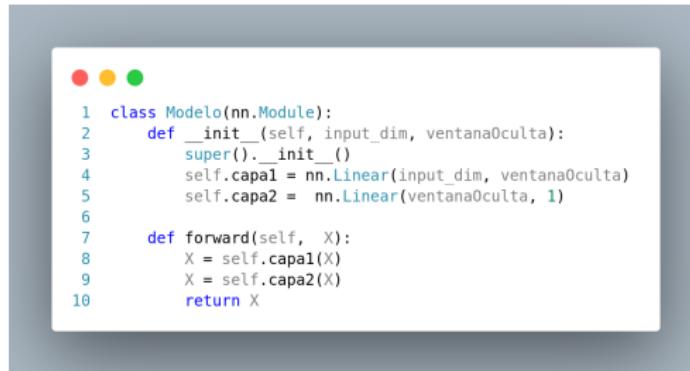


Figura: MCM98.

Implementación del MCB

- ▶ Función de pérdida: BCEWithLogistLoss
- ▶ Algoritmo de optimización: AdamW



```
● ● ●
1 class Modelo(nn.Module):
2     def __init__(self, input_dim, ventanaOculto):
3         super().__init__()
4         self.capa1 = nn.Linear(input_dim, ventanaOculto)
5         self.capa2 = nn.Linear(ventanaOculto, 1)
6
7     def forward(self, X):
8         X = self.capa1(X)
9         X = self.capa2(X)
10        return X
```

Figura: Definición de la clase del modelo de clasificación binaria.

Hiperparámetro	Posibles valores
<i>Batch size</i>	[2000, 10000, 15000, 20000]
<i>Learning rate</i>	$[10^{-2}, 10^{-3}, 10^{-4}]$
Épocas	[10, 20, 30]

Cuadro: Valores de los hiperparámetros utilizados en los experimentos del modelo de clasificación binaria.

Implementación del MCB

- ▶ Función de pérdida: CrossEntropyLoss
- ▶ Algoritmo de optimización: AdamW



```
● ● ●
1 class ModeloMulticlasse(nn.Module):
2     def __init__(self, input_dim, ventanaOculta, numClases):
3         super().__init__()
4         self.capa1 = nn.Linear(input_dim, ventanaOculta)
5         self.bn1 = nn.BatchNorm1d(ventanaOculta, momentum=0.01)
6         self.capa2 = nn.Linear(ventanaOculta, numClases)
7
8     def forward(self, X):
9         X = torch.relu(self.bn1(self.capa1(X)))
10        X = self.capa2(X)
11        return X
```

Figura: Definición de la clase del modelo de clasificación multiclas.

Hiperparámetro	Posibles valores
<i>Batch size</i>	[32, 64, 128, 256, 512]
<i>Learning rate</i>	[10^{-2} , 10^{-3} , 10^{-4} , 10^{-5}]
Épocas	[30, 50, 80, 100]

Cuadro: Valores de los hiperparámetros utilizados en los experimentos del modelo de clasificación multiclas.

Selección de los mejores MCB

Los mejores resultados del MCB los obtuvo la arquitectura con el doble de neuronas en su capa oculta de atributos de entrada tiene el modelo.

Posicion_EXP	1º-MCB98	2º-MCB98	3º-MCB98	4º-MCB98	5º-MCB98
batch_size	20000	10000	15000	15000	10000
epoches	10	30	30	10	20
learning_rate	10 ⁻²	10 ⁻³	10 ⁻³	10 ⁻²	10 ⁻³
avg_f1	0.998124	0.997941	0.997771	0.998015	0.997730
avg_fn	18.4	21.6	22	22.4	22.6
avg_fp	36.8	39	43.6	36	44.2
avg_precision	0.997501	0.997351	0.997039	0.997554	0.996999
avg_recall	0.998749	0.998531	0.998504	0.998477	0.998463
avg_roc_auc	0.999781	0.999777	0.999776	0.999773	0.999777
avg_tn	344127.4	344125.2	344120.6	344128.2	344120
avg_tp	14686.2	14683	14682.6	14682.2	14682

Cuadro: Mejores cinco configuraciones para el MCM98.

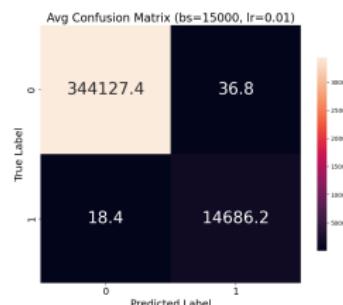


Figura: Matriz de confusión 1º MCM98.

Mejores 5 configuraciones del MCB

En este modelo no se observa que la arquitectura influya especialmente en los resultados obtenidos.

batch_size	epochs	hidden_size	learning_rate	avg_recall
20000	10	98	0.01	0.9987486972
20000	10	49	0.01	0.9986670886
10000	30	49	0.001	0.9986534868
15000	10	25	0.01	0.9986126862
15000	10	49	0.01	0.9985718792

Figura: Mejores cinco configuraciones de hiperparámetros del modelo de clasificación binaria.

Selección de los mejores MCB

Al igual que en el MCB, los mejores resultados del MCM se han obtenido en la arquitectura con el doble de neuronas en su capa oculta de atributos de entrada tiene el modelo.

Posicion_EXP	1º-MCM98	2º-MCM98	3º-MCM98	4º-MCM98	5º-MCM98
batch_size	256	256	512	64	256
epochs	100	80	100	80	50
learning_rate	10^{-3}	10^{-3}	10^{-2}	10^{-3}	10^{-3}
avg_accuracy	0.569414	0.556057	0.556915	0.556969	0.562083
avg_f1_macro	0.413180	0.406773	0.388808	0.398720	0.397308
avg_f1_weighted	0.583123	0.577553	0.574200	0.571020	0.569831
avg_precision_macro	0.394898	0.385322	0.372836	0.377543	0.387029
avg_precision_weighted	0.681971	0.675326	0.674534	0.669836	0.669688
avg_recall_macro	0.577069	0.564962	0.573083	0.566046	0.550391
avg_recall_weighted	0.569414	0.556057	0.556915	0.556969	0.562083
avg_roc_auc_ovo	0.813320	0.806782	0.809286	0.812789	0.800316
avg_roc_auc_ovr	0.788041	0.784984	0.781135	0.782758	0.777422

Cuadro: Mejores cinco configuraciones para el MCM98.

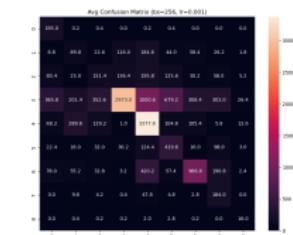


Figura: Matriz de confusión 1º MCM98.

Mejores 5 configuraciones del MCM

En el caso del MCM, la arquitectura MCM98 ha obtenido en la fase de entrenamiento unos resultados muy superiores a las otras dos arquitecturas desarrolladas.

batch_size	epochs	hidden_size	learning_rate	avg_f1_weighted
256	100	98	0.001	0.5831225815
256	80	98	0.001	0.5775525405
512	100	98	0.01	0.5741997027
64	80	98	0.001	0.5710195986
256	50	98	0.001	0.5698308505
256	50	49	0.001	0.5698160035
128	80	98	0.001	0.5669125716
128	100	98	0.001	0.5657792633
128	100	49	0.001	0.5655814612
128	80	49	0.001	0.5654025284

Figura: Mejores cinco configuraciones de hiperparámetros del modelo de clasificación multiclas.

Índice

1. Introducción

2. Planificación y costes

3. Metodología

4. Entendimiento del problema

5. Entendimiento de los datos

6. Modelado

7. Evaluación

8. Despliegue

9. Conclusiones

Índice

1. Introducción

2. Planificación y costes

3. Metodología

4. Entendimiento del problema

5. Entendimiento de los datos

6. Modelado

7. Evaluación

8. Despliegue

9. Conclusiones

Índice

1. Introducción

2. Planificación y costes

3. Metodología

4. Entendimiento del problema

5. Entendimiento de los datos

6. Modelado

7. Evaluación

8. Despliegue

9. Conclusiones