



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**Aplicación del Aprendizaje  
Semisupervisado en el  
descubrimiento de ataques a  
Sistemas de Recomendación**



Presentado por Patricia Hernando Fernández  
en Universidad de Burgos — 15 de octubre  
de 2022

Tutor: Álvaro Arnaiz González







UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



D. Álgvar Arnaiz González, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Expone:

Que la alumna D.<sup>a</sup> Patricia Hernando Fernández, con DNI 71362977A, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado «Aplicación del Aprendizaje Semisupervisado en el descubrimiento de ataques a Sistemas de Recomendación».

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 15 de octubre de 2022

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. Álgvar Arnaiz González

D. nombre co-tutor





## Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

## Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

## **Abstract**

A **brief** presentation of the topic addressed in the project.

## **Keywords**

keywords separated by commas.



---

# Índice general

---

<b>Índice general</b>	<b>iii</b>
<b>Índice de figuras</b>	<b>v</b>
<b>Índice de tablas</b>	<b>vi</b>
<b>Introducción</b>	<b>1</b>
1.1. Preámbulo . . . . .	1
<b>Objetivos del proyecto</b>	<b>3</b>
<b>Conceptos teóricos</b>	<b>5</b>
3.1. Apendizaje automático . . . . .	5
3.2. Aprendizaje semisupervisado . . . . .	5
3.3. Ataques a sistemas de recomendación . . . . .	7
<b>Técnicas y herramientas</b>	<b>13</b>
<b>Aspectos relevantes del desarrollo del proyecto</b>	<b>15</b>
<b>Trabajos relacionados</b>	<b>17</b>
6.1. TO-DO: dividir en subsecciones . . . . .	17
6.2. Aprendizaje semisupervisado aplicado a la detección de ataques en sistemas de recomendación . . . . .	17
6.3. Ataques en sistemas de recomendación . . . . .	17
<b>Conclusiones y Líneas de trabajo futuras</b>	<b>19</b>

<b>Bibliografía</b>
---------------------

21

---

# Índice de figuras

---

3.1. Clasificación sugerida por [1] . . . . .	6
---	---

---

# Índice de tablas

---

3.1. Descripción de los ataques básicos . . . . .	9
3.2. Descripción de los ataques con poco conocimiento del sistema. .	9
3.3. Descripción de las estrategias de ofuscación . . . . .	10

---

# Introducción

---

## 1.1. Preámbulo

A diferencia de unas décadas atrás, la sociedad actual está gobernada por los datos. La transición a la era de la información puede ser compleja para determinados colectivos y, consecuentemente, diversos sistemas auxiliares han sido desarrollados con el fin de resumir información y facilitar la toma de decisiones. Entre ellos se encuentran los sistemas de recomendación, que son herramientas que pretenden realizar sugerencias de objetos que pueden resultar interesantes para un determinado perfil.

Económicamente, este tipo de algoritmo es un claro objeto de interés, puesto que puede influir en la toma de decisiones de los compradores y hacer que se inclinen por un determinado producto (por ejemplo, el que tenga una mejor valoración). Los atacantes conocen esta situación y manipulan estas herramientas mediante el uso de perfiles falsos con el fin de beneficiar sus productos o perjudicar los de la competencia.

Este proyecto de investigación pretende explorar cómo el aprendizaje semisupervisado puede ayudar a detectar los ataques a sistemas de recomendación, diferenciando entre perfiles genuinos e inyectados, además de comprobar la veracidad de los planteados por otros investigadores.



---

## Objetivos del proyecto

---

Este apartado explica de forma precisa y concisa cuales son los objetivos que se persiguen con la realización del proyecto. Se puede distinguir entre los objetivos marcados por los requisitos del software a construir y los objetivos de carácter técnico que plantea a la hora de llevar a la práctica el proyecto.





---

## Conceptos teóricos

---

Se sintetizarán a continuación algunos de los conceptos teóricos más relevantes para la correcta comprensión del documento.

### 3.1. Aprendizaje automático

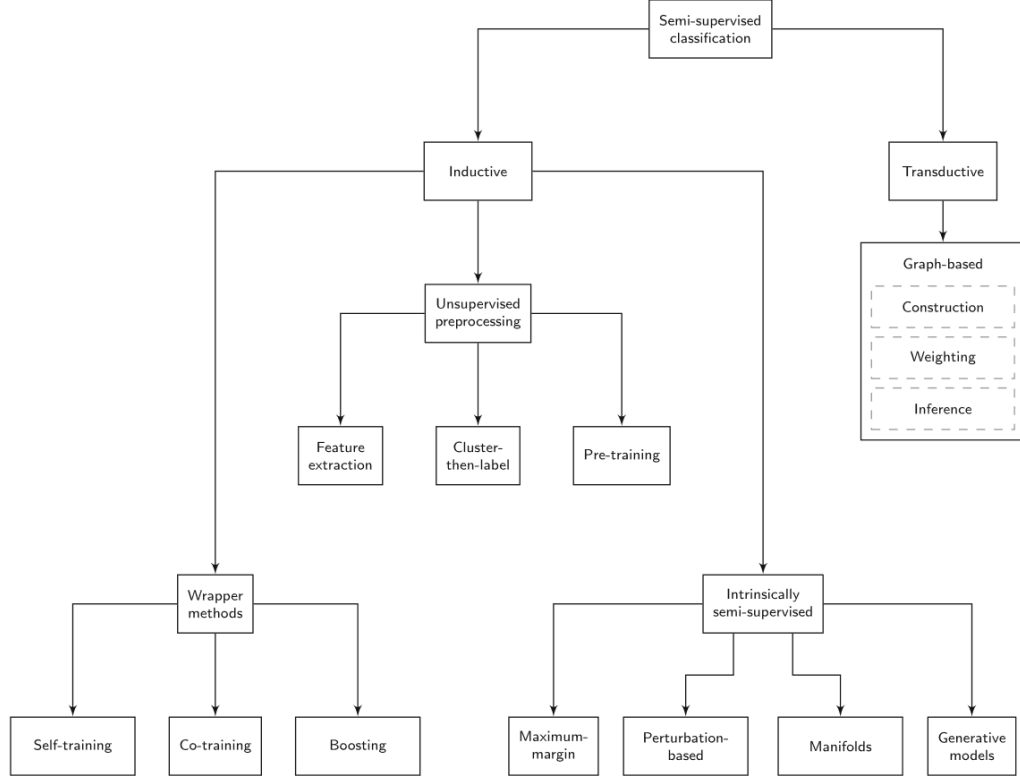
Se denomina aprendizaje automático a aquella rama de la inteligencia artificial cuyo objetivo es desarrollar métodos que permitan que un algoritmo mejore su rendimiento mediante la experiencia y procesamiento de datos. Consecuentemente, los modelos entrenados realizarán predicciones cada vez más precisas como resultado del algoritmo implementado.

Dentro del aprendizaje automático se diferencian tres grandes grupos en función del tipo de entrada que sea consumida: el aprendizaje supervisado (datos etiquetados), el no supervisado (datos no etiquetados) y el semisupervisado (datos etiquetados y no etiquetados), siendo esta última categoría objeto de estudio en este proyecto de investigación.

### 3.2. Aprendizaje semisupervisado

Como se ha mencionado anteriormente, se denomina aprendizaje semisupervisado a aquel conjunto de algoritmos que utiliza datos etiquetados y no etiquetados para realizar tareas de aprendizaje. Inicialmente, se pueden diferenciar dos categorías[1]: los métodos inductivos, cuyo objetivo principal es construir un clasificador que genere predicciones para cualquier entrada y los métodos transductivos, cuyo poder de predicción está limitado a los objetos utilizados en la fase de entrenamiento.

Figura 3.1: Clasificación sugerida por [1]



Prescindiendo de los métodos transductivos por ser menos versátiles y útiles en nuestro propósito, los métodos inductivos se subdividen en tres grupos [1]: *wrapper methods* (o métodos de envoltura), *unsupervised preprocessing* y *intrinsically semi-supervised*, siendo materia de estudio los métodos de envoltura.

## Métodos de envoltura

Estos modelos utilizan uno o más clasificadores que son entrenados iterativamente con los datos etiquetados de entrada, además de con datos pseudoetiquetados. Se denomina pseudoetiquetado a aquellos datos que inicialmente no estaban etiquetados, pero acabaron estándolo por iteraciones previas de los clasificadores.

Consecuentemente, el procedimiento consta de dos fases que se repiten en cada iteración: el entrenamiento y el pseudoetiquetado. Durante el entrenamiento, los clasificadores se alimentan de datos etiquetados (o pseudoetiquetados). En la fase de pseudoetiquetado, se utilizan datos no

etiquetados para que sean procesados por los clasificadores previamente entrenados.

Dentro de esta categoría, se pueden diferenciar tres grandes grupos: *self-training*, que utilizan únicamente un clasificador, *co-training*, que utilizan más de uno y los *pseudo-labelled boosting methods*, que construyen clasificadores individuales que se alimentan de las predicciones más fiables. Se estudiará más en profundidad los métodos *co-training*.

### Co-training y Co-forest

En estos algoritmos, varios clasificadores son entrenados iterativamente utilizando datos etiquetados y añadiendo las predicciones (resultados) más certeras al conjunto para ser utilizadas en las siguientes iteraciones. Para que los clasificadores sean capaces de generar información distinta, generalmente se divide el conjunto de entrada según alguna característica (no siendo estrictamente necesario).

El llamado *co-forest*, es un modelo dentro del *co-training*. En su desarrollo, se utilizan árboles de decisión (a mayor número mejor resultado), que son entrenados utilizando los datos etiquetados. En cada iteración, además, se añade al conjunto de datos nuevos elementos pseudoetiquetados. Estos elementos son el resultado de los elementos comunes (nuevas etiquetas) del resto de árboles en la fase anterior, y se usan durante una fase de entrenamiento. Sin embargo, se eliminan una vez se ha completado (la siguiente iteración se realiza inicialmente sólo con los datos etiquetados, etc.), consiguiendo así resultados certeros.

## 3.3. Ataques a sistemas de recomendación

Los ataques a los sistemas de recomendación (generalmente denominados *shilling attacks* o *profile injection attack*) tienen como objetivo manipular las sugerencias que propone un determinado algoritmo para conseguir que un cliente se incline hacia un elemento deseado. Esta alteración del sistema se consigue inyectando perfiles falsos.

Múltiples estudios se han centrado en formalizar las características de estos ataques con el fin de detectarlos. Entre ellas se encuentran:

- **Intención:** normalmente, se pretende manipular la opinión general acerca de un elemento (ya sea para bien o para mal). Según el objetivo

se pueden diferenciar dos tipos de ataques: *push attacks*, que pretenden hacer un objeto más atractivo o *nuck attacks*, cuya intención es la contraria. En caso de que el atacante no busque alterar la opinión acerca de un producto sino restar credibilidad a un sistema (mediante valoraciones aleatorias), se habla de *random vandalism*.

- **Fuerza:** la calidad de los ataques se mide teniendo en cuenta el **tamaño del relleno** (número de valoraciones asignadas a un perfil atacante, que suele rondar entre el 1 y el 20 % del total de los ítems [3]) y el **tamaño del ataque** (número de perfiles inyectados en el sistema, rondando entre el 1 y el 15 %).
- **Coste:** se distinguen dos tipos: *knowledge-cost*, que hace referencia al coste de construir perfiles y *deployment-cost*, que es el número de perfiles que se deben inyectar para conseguir un ataque efectivo.

## Tipos de ataques

En la actualidad se distinguen multitud de ataques distintos. Con el fin de formalizarlos matemáticamente, se han establecido ciertos conjuntos de interés dependiendo de los ítems que contengan [7].

- $I_S$ : conjunto de ítems seleccionados para recibir un tratamiento especial (puede ser vacío).
- $I_F$ : conjunto de ítems seleccionados para «rellenar».
- $I_0$ : conjunto de ítems pertenecientes al sistema de recomendación sin valorar.
- $I_t$ : conjunto de ítems objetivo.

## Ataques básicos

Se distinguen dos tipos: *random attack* y *average attack*. Ambos tienen parámetros y características muy similares como se muestra en la tabla 3.1. La principal diferencia reside en que el *average attack* es mucho más potente debido a que cuenta con mayor información acerca del sistema: las valoraciones a los ítems de relleno siguen una distribución  $\mathcal{N}(\mu_i, \sigma_i)$ , en lugar de  $\mathcal{N}(\mu, \sigma)$ . Es decir, la valoración para un determinado ítem se adecúa a la distribución concreta de ese ítem en lugar de a la de todo el *dataset*.

Modelo	$I_S$ :	Valoración $I_F$ :	$I_0$ :	Valoración $I_t$ :
Random	$\emptyset$	Aleatoria siguiendo una distribución normal definida por todas las valoraciones para todos los ítems del sistema $\mathcal{N}(\mu, \sigma)$ .	$\emptyset$	máxima o mínima
Average	$\emptyset$	Aleatoria siguiendo una distribución normal definida por las otras valoraciones para ese ítem en concreto $\mathcal{N}(\mu_i, \sigma_i)$ .	$\emptyset$	máxima o mínima

Tabla 3.1: Descripción de los ataques básicos

Modelo	$I_S$ :	Valoración $I_F$ :	$I_0$ :	Valoración $I_t$ :
Bandwagon (average)	Ítems populares (valoración máxima) o ítems desfavorecidos (puntuación mínima) (reverse)	Aleatoria siguiendo una distribución normal definida por las otras valoraciones para ese ítem en concreto $\mathcal{N}(\mu_i, \sigma_i)$ .	$\emptyset$	máxima o mínima (reverse)
Bandwagon (random)	Ítems populares (valoración máxima) o ítems desfavorecidos (puntuación mínima) (reverse)	Aleatoria siguiendo una distribución normal definida por todas las valoraciones para todos los ítems del sistema $\mathcal{N}(\mu, \sigma)$ .	$\emptyset$	máxima o mínima (reverse)

Tabla 3.2: Descripción de los ataques con poco conocimiento del sistema.

### Ataques con poco conocimiento del sistema.

Los más populares son *bandwagon attack* (o *popular attack*) y *segment attack*.

La principal característica del *bandwagon attack* es que el conjunto  $I_S$  ya no está vacío, sino que contiene algunos de los ítems más populares de la base de datos. Estos ítems recibirán también la máxima puntuación posible, de forma que ya no sólo se puntúa el conjunto objetivo. Existe una variante de este ataque llamada *reverse bandwagon attack*, cuyo objetivo es hacer *nuke*. De esta forma,  $I_S$  contiene los ítems menos populares y reciben la puntuación mínima (junto con  $I_t$ ).

En el *segment attack*, se realiza un pequeño «estudio de mercado» y se introduce en  $I_S$  ítems en los que estaría interesado un usuario que fuese a valorar también  $I_t$  (de forma que el ataque es más «realista»).

Modelo	Estrategia de ofuscación
Noise Injection	$\forall i \in I_F \cup I_S : R_i = r_i + \text{aleatorio} * \alpha$
User Shifting	$\forall i \in I_F \cup I_S : R_i = r_i + \text{shift}$
Target Shifting	$\forall i \in I_F \cup I_S : R_i = r_i; I_T : r_{\max} - 1 \text{ or } r_{\min} + 1$
AOP	$I_F \text{ escogido de los } \text{top } \text{items más populares}$

Tabla 3.3: Descripción de las estrategias de ofuscación

### Ataques con gran conocimiento del sistema

Este tipo de ataques resulta menos relevante que los anteriores debido a la dificultad de su ejecución. En la mayoría de los casos, se necesita una gran cantidad de información, siendo poco realista que se de una situación de estas características «en la vida real».

Por ejemplo, el llamado *perfect knowledge attack* basa su efectividad en reproducir la distribución exacta de la base de datos real (exceptuando los ítems objetivos). El *sampling attack* construye los perfiles a inyectar basándose en una muestra de perfiles reales.

Como se puede intuir, conocer datos estadísticos exactos sobre una base de datos o metadatos asociados a perfiles de usuarios es poco realista (cada vez menos debido a las mayores medidas de seguridad) y por lo tanto estos ataques resultan meramente teóricos.

### Ataques ofuscados

Los ataques ofuscados se basan en intentar «camuflar» los perfiles inyectados haciéndolos pasar por reales.

El ataque de *noise injection* introduce a los conjuntos  $I_S$  e  $I_F$  un «ruido» aleatorio (aunque sigue una distribución Gaussiana) multiplicado por una constante  $\alpha$ . *User shifting* sigue la misma estrategia, sólo que el ruido esta vez es un número aleatorio normalmente distribuido. *Target shifting* incrementa (o decrementa) en una unidad la valoración de  $I_t$  con el fin de crear diferencias entre los ataques y el *Average over popular items (AOP)* pretende ofuscar el *average attack* cambiando la forma de selección de  $I_F$  (en lugar de seleccionar los ítems del conjunto total de la colección, se seleccionan los X % ítems más populares).

**Otros tipos de ataques**

Además de los ataques anteriormente mencionados, existen otros con objetivos más diversos o estrategias distintas. El anteriormente mencionado *random vandalism* pertenece a esta categoría. Se pueden distinguir, además, ataques basados en copiar comportamientos de usuarios influyentes (modelo PUA) o ítems poderosos (modelo *PIA*). Sin embargo, son menos abundantes.





---

## Técnicas y herramientas

---

Esta parte de la memoria tiene como objetivo presentar las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto. Si se han estudiado diferentes alternativas de metodologías, herramientas, bibliotecas se puede hacer un resumen de los aspectos más destacados de cada alternativa, incluyendo comparativas entre las distintas opciones y una justificación de las elecciones realizadas. No se pretende que este apartado se convierta en un capítulo de un libro dedicado a cada una de las alternativas, sino comentar los aspectos más destacados de cada opción, con un repaso somero a los fundamentos esenciales y referencias bibliográficas para que el lector pueda ampliar su conocimiento sobre el tema.



---

## Aspectos relevantes del desarrollo del proyecto

---

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros<sup>3</sup>, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.



---

## Trabajos relacionados

---

Dentro de este proyecto se pueden diferenciar distintas líneas de investigación:

### 6.1. TO-DO: dividir en subsecciones

### 6.2. Aprendizaje semisupervisado aplicado a la detección de ataques en sistemas de recomendación

En este caso, el artículo fundamental es [7]. En este documento, se propone un método de detección basado en Co-Forest y se producen distintas comparativas con otros algoritmos para comprobar su eficacia, consiguiendo unos resultados muy aceptables. También es muy relevante citar el trabajo de [6], puesto que propone una aproximación Naive Bayes para separar perfiles de atacantes de perfiles genuinos y además propone los tipos de datasets que son utilizados posteriormente por Zhou y Duan (Amazon, Netflix y MovieLens).

### 6.3. Ataques en sistemas de recomendación

La importancia de proteger los sistemas de recomendación ha sido contemplada desde principio de siglo, siendo común la proposición de otros tipos de aprendizaje para detectar los ataques.

Respecto a la descripción de los tipos de intrusión, la correcta definición formal (matemática) de sus parámetros y una recopilación de la gran mayoría de ataques existentes, es fundamental referenciar el artículo de [3]. Previo a este documento, también es relevante contemplar otros trabajos, como la conferencia de [2], donde se propone utilizar como datasets los conjuntos de películas o el paper de [4], que define los modelos de construcciones en base a conocimiento del sistema y pone a prueba la robustez de los recomendadores evaluando su estabilidad y precisión ante la presencia de perfiles inyectados (análisis matemático muy completo).

---

## **Conclusiones y Líneas de trabajo futuras**

---

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.





---

## Bibliografía

---

Jesper Engelen and Holger Hoos. A survey on semi-supervised learning. *Machine Learning*, 109, 02 2020.

Shyong K. Lam and John Riedl. Shilling recommender systems for fun and profit. page 393 – 402, 2004. Cited by: 439.

Si Mingdan and Qingshan Li. Shilling attacks against collaborative recommender systems: a review. *Artificial Intelligence Review*, 53, 01 2020.

Michael O’Mahony, Neil Hurley, Nicholas Kushmerick, and Guénolé Silvestre. Collaborative recommendation: A robustness analysis. *ACM Trans. Internet Technol.*, 4(4):344–377, nov 2004.

Isaac Triguero, Salvador García, and Francisco Herrera. Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study. *Knowledge and Information Systems*, 42, 02 2015.

Zhiang Wu, Junjie Wu, Jie Cao, and Dacheng Tao. Hysad: A semi-supervised hybrid shilling attack detector for trustworthy product recommendation. page 985–993, 2012.

Quanqiang Zhou and Liangliang Duan. Semi-supervised recommendation attack detection based on co-forest. *Comput. Secur.*, 109(C), oct 2021.