Universidad Nacional
de Educación a Distancia

Escuela Técnica Superior de Ingeniería Informática

Metaanálisis de la aplicación de aprendizaje automático en la detección de malware

Daniel González Herrera

Director: Emilio Letón Molina
Co-director: Jorge Pérez Martín

Trabajo de Fin de Máster

Máster Universitario
en Ingeniería y Ciencia de Datos

Septiembre 2023

Agradecimientos

Me gustaría expresar mi más profunda gratitud a todas las personas que me han brindado su apoyo incondicional y han contribuido de manera fundamental a la realización de este trabajo, muy en especial a las siguientes:

- A mis tutores, Emilio Letón y Jorge Pérez, os estoy enormemente agradecido por vuestra orientación experta, vuestra paciencia y por la cantidad de tiempo que habéis dedicado a revisiones, reuniones y dudas a lo largo de todo este proceso. Vuestro conocimiento, sugerencias y comentarios han enriquecido significativamente este trabajo.
- A Antonio por tu apoyo constante y tus palabras de aliento en los momentos más difíciles y estresantes. Valoro especialmente tu compresión y disposición a sacrificar el poco tiempo juntos del que hemos dispuesto.
- A mi madre, porque desde pequeño me enseñaste a perseverar y a no rendirme. Y porque sé que eres la persona que siempre está más orgullosa de los distintos logros académicos que he conseguido.

Daniel González Herrera
Málaga, Septiembre 2023

Resumen

El aumento de las amenazas cibernéticas y la evolución constante de las técnicas de ataque han impulsado la necesidad de sistemas de detección más robustos y eficientes. En este contexto, el aprendizaje profundo ha emergido como un enfoque prometedor, aprovechando su capacidad para extraer patrones y características complejas a partir de grandes volúmenes de datos.

El objetivo del presente trabajo consiste en evaluar si la aplicación de técnicas de aprendizaje profundo, en el ámbito de la detección de *malware*, ofrece resultados positivos para su aplicación en este entorno. Ello se aborda mediante una revisión exhaustiva y sistemática de los estudios que aplican cualesquiera de estas técnicas (CNN, RNN, *Autoencoders*, ...) para detección de *malware*. El método utilizado consiste en la realización de un metaanálisis detallado de estas investigaciones, siguiendo la guía PRISMA, y teniendo en cuenta algunas características para la estratificación de dicho metaanálisis, como son la plataforma sobre la que se recibe el ataque y el tipo de análisis llevado a cabo.

Los resultados obtenidos en el desarrollo del metaanálisis ofrecen una alta heterogeneidad, lo que no permite poder asegurar que sus conclusiones numéricas sean correctas. Sin embargo, el análisis crítico de todo el proceso ofrece, sin lugar a dudas, una lectura positiva en la aplicación de las técnicas de aprendizaje profundo para la detección de *malware*.

Palabras clave: *malware*, aprendizaje profundo, metaanálisis

Abstract

The increase in cyber threats and the constant evolution of attack techniques have driven the need for more robust and efficient detection systems. In this context, deep learning has emerged as a promising approach, harnessing its ability to extract complex patterns and features from large volumes of data.

The objective of this study is to evaluate whether the application of deep learning techniques, in the field of malware detection, yields positive results for implementation in this environment. This is addressed through a comprehensive and systematic review of studies that apply any of these techniques (CNN, RNN, Autoencoders, etc.) for malware detection. The method used involves conducting a detailed meta-analysis of these research studies, following the PRISMA guidelines, and considering certain characteristics for the stratification of this meta-analysis, such as the platform on which the attack is received and the type of analysis carried out.

The results obtained from the development of the meta-analysis exhibit high heterogeneity, which prevents us from guaranteeing the correctness of its numerical conclusions. However, the critical analysis of the entire process undeniably provides a positive outlook on the application of deep learning techniques for malware detection.

Keywords: malware, deep learning, meta-analysis

Glosario

AE *Autoencoder* o Autocodificador.

AES Acrónimo de *Advanced Encryption Standard* o Estándar de Cifrado Avanzado. Se trata de un algoritmo criptográfico de clave privada o cifrado simétrico..

ANOVA *Analysis of Variance* o Análisis de la Varianza.

análisis dinámico Es un tipo de análisis de *malware* en el que se ejecuta dicho *malware* en un entorno controlado para observar su comportamiento.

análisis estático Es un tipo de análisis de *malware* en el que se toman características del mismo sin llegar a ejecutarlo.

análisis híbrido Es un tipo de análisis de *malware* que combina aspectos de los análisis estático y dinámico.

aprendizaje profundo (*deep learning*) Es un tipo de aprendizaje automático en el que se utilizan redes neuronales con múltiples capas ocultas para extraer patrones complejos.

ARAE *Attention Recurrent Autoencoder* o Autocodificador Recurrrente Atencional.

ataque adversario Es un conjunto de técnicas que manipulan los datos que se le suministra a una red neuronal para conseguir que clasifique ese dato de forma errónea.

ataque de día cero Es un ataque *malware* que se lanza sobre una vulnerabilidad que no ha sido detectada por los desarrolladores y por tanto no dispone de parche corrector.

CNN *Convolutional Neural Network* o Red Neuronal Convolucional.

CRNN *Convolutional Recurrent Neural Network* o Red Neuronal Convolucional Recurrente, consiste la combinación de un Autoencoder convolucional (OCAE) y una Red Neuronal Recurrente Profunda (DRNN).

DBN *Deep Belief Network* o Red de Creencias Profunda. Es una combinación de múltiples redes simples de tipo RBM.

diagrama de efectos (*forest plot*) Es una herramienta gráfica que permite evaluar los efectos individuales y conjuntos de los estudios en un metaanálisis.

diagrama de embudo (*funnel plot*) Es una herramienta gráfica para visualizar el sesgo de publicación mediante la dispersión de los resultados.

DNN *Deep Neural Network* o Red Neuronal Profunda.

DR Diferencia de riesgos o *Risk Difference*.

especificidad Razón entre los verdaderos negativos y los negativos detectados (verdaderos negativos + falsos positivos).

FCSCNN *Feature centralized siamese convolutional neural network* o Red Neuronal Convolucional Siamesa Centralizada en Características.

FEM *Fixed Effects Model* o Modelo de efectos fijos.

FT *Freeman-Tukey Transformation* o Transformación Doble Arcoseno.

GCN *Graph Convolutional Network* o Red Convolucional de Grafos.

GLMM *Generalized Linear Mixed Model* o Modelo Lineal Mixto Generalizado.

GNN *Graph Neural Network* o Red Neuronal de Grafos.

heterogeneidad Es la medida de cómo de distintos son los resultados de los estudios usados en el metaanálisis.

IdC Internet de las Cosas o *Internet of things (IoT)*.

LSTM *Long Short Term Memory* o Memoria a Corto Plazo de Larga Duración.

malware Programa o código diseñado para infectar, dañar o acceder a sistemas informáticos.

malware metamórfico Es un tipo de *malware* que es capaz de reescribir completamente el código que lo forma para evitar su detección.

malware oligomórfico Es un tipo de *malware* que es capaz de generar unas cientos de versiones ligeramente diferentes de la original pero manteniendo su funcionalidad.

malware polimórfico Es un tipo de *malware* que es capaz de generar millones de versiones diferentes de la original pero manteniendo su funcionalidad.

medida de efecto También llamado tamaño del efecto, es la relación o medida de fuerza existente entre dos variables de una población o muestra.

metaanálisis Técnicas estadísticas que permiten combinar los resultados de estudios que investigan una misma cuestión para obtener una estimación más precisa de la evidencia.

metaanálisis de proporciones Es un tipo de metaanálisis que estima una proporción general de un resultado específico.

metaanálisis tradicional Es un tipo de metaanálisis que analiza el efecto de una variable independiente en un resultado final de modo que pueden discernirse resultados positivos y negativos claramente.

MLP *Multilayer Perceptron* o Perceptrón Multicapa.

ofuscación Es una técnica que modifica la estructura de un programa o código con el fin de hacerlo más difícil de interpretar y comprender sus objetivos.

OR *Odds ratio* o Razón de Disparidades.

PRISMA *Preferred Reporting Items for Systematic Reviews and Meta-Analyses* o Elementos de Informe Preferidos para Revisiones Sistemáticas y Metaanálisis. Es una Metodología para la realización de revisiones sistemáticas y metaanálisis con el objeto de mejorar su transparencia y calidad.

RBM *Restricted Boltzman Machine* o Máquina de Boltzman Restringidas.

REM *Random Effects Model* o Modelo de efectos aleatorios.

revisión sistemática Artículo que realiza una revisión de todos los estudios de un determinado tema siguiendo una metodología transparente que permite su replicación.

RNN *Recurrent Neural Network* o Red Neuronal Recurrente.

RR *Risk Ratio* o Riesgo Relativo.

RSA Acrónimo de *Rivest, Shamir y Adleman*. Se trata de un algoritmo criptográfico de clave pública que fundamenta su seguridad en la complejidad de la factorización de números enteros muy grandes.

sensibilidad Razón entre los verdaderos positivos y los positivos detectados (verdaderos positivos + falsos negativos).

sesgo de publicación Es la tendencia hacia un resultado que no es el real debido a la falta de publicación de estudios sin resultados significativos.

SVM *Support Vector Machine* o Máquinas de Vectores de Soporte.

TCN *Temporal Convolutional Network* o Red Convolucional Temporal.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Los desarrollos tecnológicos de las últimas décadas han permitido que la sociedad esté cada vez más interconectada. Asimismo, el fácil acceso a las redes y, en particular, a Internet ha propiciado un auge del consumo y compartición de información digitalizada en un corto intervalo de tiempo. Sin ir más lejos, se estima que más de 5 mil millones de personas en el mundo (alrededor del 66 % de la población mundial indicada en por el Banco Mundial[1],como se puede ver en la Tabla 1.1) han usado Internet a lo largo del año 2022, un número que aumenta año tras año. Además, el objetivo es que, en el año 2030, el 100 % de la población tenga acceso a Internet (International Telecommunication Union (ITU) [2023]).

Tabla 1.1: Evolución del acceso a Internet en los últimos 5 años. Datos de usuarios y habitantes en miles de millones (Fuentes: International Telecommunication Union (ITU) [2023], Banco Mundial)

Año	**Nº de usuarios de Internet**	**Nº de habitantes**	**Internautas/Habitante**
2018	3,7	7,66	48,30 %
2019	4,2	7,74	54,26 %
2020	4,7	7,82	60,10 %
2021	4,9	7,89	62,10 %
2022	5,3	7,95	66,66 %

Esta situación ha ofrecido a los ciberdelincuentes una gran oportunidad para expandir sus ciberataques hacia un mayor número de usuarios y empresas. Según AAG IT [2023], de media ocurre un cibercrimen cada 37 segundos y, durante 2022, se filtró información privada de 2 internautas por segundo. Los ciberataques, ocasionan un enorme perjuicio tanto a consumidores finales como a empresas e instituciones. No en vano, se calcula que el coste económico a nivel mundial debido al cibercrimen es de unos 7 billones de dólares y se estima que ese coste aumentará hasta los 10,5 billones de dólares en 2025 (AAG IT [2023]).

Existen diversos tipos de ciberataques, cada uno de ellos necesita de la aplicación de una estrategia defensiva distinta. Entre los ciberataques más comunes se encuentran: el

[1]https://datos.bancomundial.org/indicator/SP.POP.TOTL

phising, que consiste en la suplantación de identidad de una entidad legítima, como un banco, mediante la creación de un sitio web o un correo electrónico que haga creer al usuario que está tratando con la empresa en cuestión y así extraerle información confidencial; y los ataques por *malware*, que consisten en la instalación de software malicioso en algún sistema con el objetivo de controlarlo, robar o eliminar información, propagar virus o *ransomware* o enviar *spam*.

La ciberseguridad es el conjunto de estrategias, técnicas, procedimientos y herramientas que permiten evadir o combatir los ciberataques. La utilización de antivirus, la aplicación de *firewalls*, la actualización de software mediante parches o el análisis del tráfico de red, conjuntamente con la correcta educación y concienciación del usuario son algunas de sus principales medidas.

Según AV-Test Institute [2023], existen más de 1050 millones de aplicaciones maliciosas clasificadas actualmente y se registran alrededor de 450.000 nuevas cada día. Desde el nacimiento del primer *malware*, allá por 1970, los ataques por *malware* han evolucionado y continúan haciéndolo constantemente ya que los ciberdelincuentes siempre están desarrollando nuevas formas de saltarse las medidas de seguridad o aprovechar las brechas de los distintos sistemas en su favor. Aunque Windows sigue siendo el sistema preferido por los atacantes, el surgimiento de nuevos dispositivos, la aparición del Internet de las Cosas (IdC) y el desarrollo de sistemas operativos como Android también han colaborado en la expansión de este tipo de ataques.

Por ello, es muy necesario que las técnicas tanto defensivas como proactivas de la ciberseguridad estén vigilantes y en constante desarrollo, utilizando todas las mejoras tecnológicas que tengan a su alcance para contrarrestar estos ataques. Durante los años se han ido implementando diferentes estrategias que intentan automatizar la captura del *malware* antes de su instalación o ejecución en un dispositivo: métodos basados en la firma, métodos estáticos, dinámicos o híbridos son, como se verá más adelante, algunas de estas técnicas. Además, las técnicas de aprendizaje automático han añadido mucho valor a estas formas de detección.

1.1. Motivación

En la última década, las técnicas de aprendizaje automático y, en particular, las de aprendizaje profundo han tenido un desarrollo espectacular. Tanto es así, que se están aplicando a tareas tan diversas como: diagnósticos médicos basados en imágenes (Bakator and Radosav [2018]), conducción autónoma (Muhammad et al. [2021]), marketing (Jiang [2021]), reconocimiento de voz (Chandolikar et al. [2022]), procesamiento de lenguaje natural(Otter et al. [2021]), o inteligencias artificiales que generan imágenes o chats (He and Deng [2017]).

Las técnicas de detección de *malware* no iban a ser menos. En los últimos cinco años se han llevado a cabo múltiples estudios, como se verá en capítulos siguientes, en los que se aplican técnicas de aprendizaje profundo en distintas fases de la detección para intentar

obtener mejores resultados a la hora de frenar los ataques *malware* (Berman et al. [2019]).

Por otra parte, el punto central de este trabajo va a ser la realización de una revisión sistemática y de un metaanálisis con el cual se pretende analizar la situación actual basada en los estudios realizados hasta el momento. Como se explicará más adelante, el metaanálisis es una técnica estadística muy utilizada en el ámbito científico para evaluar los resultados de un grupo de estudios que atacan a un mismo objetivo dentro de una determinada temática. Así, mediante la aplicación de esta técnica estadística a una revisión sistemática de una serie de artículos, se podrá poner en relieve la utilidad del aprendizaje profundo en la detección de *malware* hoy en día.

1.2. Propuesta y objetivos

Con la realización de este Trabajo Fin de Máster se pretende analizar si las técnicas de aprendizaje profundo y las redes neuronales profundas suponen una herramienta útil en la detección de *malware*.

Para ello, se van a responder a una serie de cuestiones que van a ser marcadas como objetivos:

- **Objetivo principal**. Conocer en qué medida la aplicación de técnicas de aprendizaje profundo permiten la detección de *malware*.
- **Objetivo específico 1**. Establecer las diferencias encontradas en las plataformas analizadas (Android y Windows).
- **Objetivo específico 2**. Reconocer las divergencias existentes respecto a los tipos de análisis utilizados en la detección en cada una de las plataformas.
- **Objetivo específico 3**. Analizar si existe algún modelo de aprendizaje profundo que ofrezca mejores resultados en alguno de los contextos indicados en los objetivos anteriores.

Como técnica de resolución de estos objetivos, se propone la realización de una revisión sistemática con metaanálisis de los estudios realizados en el campo de la detección de *malware* con aprendizaje profundo, mediante la cual se analizarán algunas métricas comunes (sensibilidad y especificidad) y se tomarán en cuenta algunas características que pudieran afectar al resultado como la plataforma que recibe el ataque, el tipo de análisis usado o la técnica de aprendizaje profundo usada.

1.3. Estructura del documento

El resto del documento se articula de acuerdo a la siguiente estructura:

- El Capítulo 2, «Estado del Arte y Marco Teórico», explicará la situación actual del tema a tratar así como los conceptos principales que van a ser utilizados en el resto del documento. En particular, se va a presentar la idea del metaanálisis, sus características, sus ventajas y sus posibles problemáticas. También se expondrá la concepción de *malware*, sus tipos y técnicas de detección así como los ataques que más problemas provocan de cara a ser detectados. Por último se realizará una breve exposición de las técnicas de aprendizaje profundo usadas en la detección de *malware*.
- En el Capítulo 3, «Materiales y Métodos», se expondrán qué fuentes de información, técnicas, algoritmos y experimentos que se han utilizado para el estudio del tema.
- En el Capítulo 4, «Resultados», se presentarán los resultados de los experimentos indicados en el capítulo anterior.
- El Capítulo 5, «Discusión», responderá a los objetivos principal y específicos que se pretenden resolver con este trabajo mediante el análisis de los resultados obtenidos en el capítulo anterior.
- Por último, en el Capítulo 6, «Conclusiones y Trabajos Futuros», se realizará un breve resumen del trabajo realizado, así como los posibles trabajos futuros derivados de este.

Capítulo 2

Estado del Arte y Marco Teórico

El campo de la detección de *malware* ha experimentado una evolución significativa a lo largo de los años. Aunque históricamente las técnicas de análisis de firmas y aplicación de reglas heurísticas han sido las más utilizadas, sufren de grandes limitaciones ante ataques cada día más sofisticados.

En contraste, la evolución del aprendizaje profundo en los últimos años ha permitido revolucionar la forma en la que se aborda la detección de *malware* usando estas técnicas. Basta repasar el número de estudios realizados en los últimos años mediante una búsqueda en cualquier base de datos científica para observar esta situación.

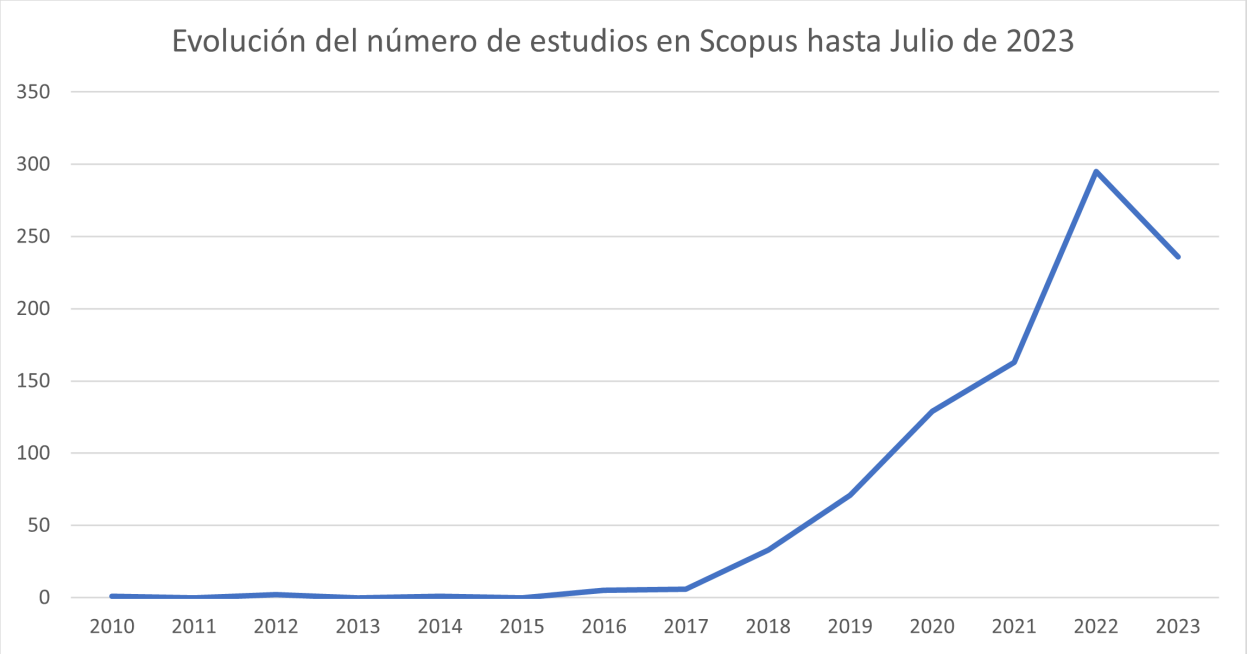

Figura 2.1: Evolución de estudios en Scopus en Julio de 2023

En la Figura 2.1 se puede contemplar que, desde el año 2018 hasta julio de 2023, el número de estudios ha crecido año a año y que a partir del año 2021 este crecimiento parece ser mucho más intenso. Esto puede ser debido principalmente a tres factores: el gran

desarrollo de los últimos años que ha experimentado el *deep learning* gracias a las mayores capacidades computacionales existentes, el tratarse de un campo de aplicación muy novedoso en el que aún hay mucha posibilidad de mejora, y a los buenos resultados que, en general, ofrecen estas técnicas en los estudios que se han ido realizando.

El metaanálisis es una herramienta muy utilizada en el campo de la epidemiología y desde la década de 1990 hasta nuestros días el crecimiento de su utilización para contrastar estudios ha sido espectacular. Pero no solo ha sido así en el campo de la epidemiología, también se ha ido aplicando a otras ciencias de forma bastante efectiva. Esta situación provoca que, en prácticamente cualquiera de los temas relevantes de la sociedad que posea un número suficiente de estudios, exista un metaanálisis realizado. Aunque es cierto que entre los estudios de aplicación de técnicas de aprendizaje profundos sobre detección de *malware* aparece alguna revisión sistemática que permite evaluar si las técnicas de aprendizaje profundo están ofreciendo buenos resultados, no es menos cierto que no existe, a fecha de julio de 2023, ningún metaanálisis que evalúe cuantitativamente el impacto real de estas técnicas.

En el contexto de la detección de *malware*, se han utilizado diferentes arquitecturas de aprendizaje profundo donde las principales son las redes neuronales convolucionales (CNN) y las redes neuronales recurrentes (RNN). Por otra parte, no todos los estudios utilizan los mismos conjuntos de datos para realizar su entrenamiento. Además, hay una amplia gama de dispositivos a la que pueden estar dirigidos los ataques. Todas estas cuestiones pueden, en cierto modo, ejercer una dificultad añadida a la realización del metaanálisis debido a la complejidad de características que hay que tener en cuenta (lo que provoca una gran heterogeneidad) y al número escaso de estudios que puede haber si se plantea una estratificación con demasiados niveles.

Un factor importante a tener en cuenta es que no todos los estudios que aparecen inicialmente en la búsqueda realizada deben ser incluidos en el metaanálisis. Se excluirán estudios por motivos cualitativos (por baja calidad, por encontrarse fuera del ámbito de la investigación, ...) o por motivos cuantitativos (tamaño muestral pequeño, ausencia de los indicadores buscados, ...).

Tras la aplicación del cribado a los estudios encontrados en las diversas búsquedas, se ha podido comprobar que existe un número suficiente como para realizar un metaanálisis de proporciones estratificado, cuestión que se abordará en los capítulos siguientes.

A continuación, se definen los principales conceptos que van a ser utilizados a lo largo del resto del trabajo.

2.1. Revisiones sistemáticas y metaanálisis

Un **artículo de revisión** es un tipo de artículo en el cual se realiza un análisis crítico de otros artículos previamente publicados acerca de una materia o temática de estudio. Hay distintas categorías de artículos de revisión siendo los más destacados: las revisiones

narrativas y las revisiones sistemáticas (Palmatier et al. [2018]).

- **Revisiones narrativas**. Describen la información publicada acerca de un determinado tema y son llevadas a cabo por expertos en dicho tema. La desventaja de este tipo de artículos es que no suelen incluir un proceso metodológico acerca de cómo han realizado la extracción de la información o qué preguntas se han llevado a cabo para filtrar los artículos que se toman en consideración, es decir, suelen tener un cierto sesgo y subjetividad.

- **Revisiones sistemáticas**. Al igual que las anteriores, recopilan y resumen un tema de estudio, pero en este caso pretenden ser más transparentes que las anteriores al disponer de un diseño previo y una metodología bien establecida y replicable. Dentro de este tipo de revisiones se distinguen a su vez dos categorías: las cualitativas y las cuantitativas. Las revisiones sistemáticas cualitativas se centran en describir las evidencias sin realizar ningún tipo de análisis estadístico (sin metaanálisis), mientras que en las cuantitativas se usan técnicas estadísticas para combinar los resultados de los distintos estudios numéricamente (con metaanálisis) y al mismo tiempo pueden incluir evidencias descriptivas (Aguilera Eguía [2014]).

Con los datos obtenidos en una revisión sistemática, se puede realizar un metaanálisis. Un **metaanálisis** es un conjunto de técnicas estadísticas que permite combinar los resultados de diferentes estudios, independientes entre sí, pero que investigan la misma cuestión con el fin de obtener una estimación más precisa de la evidencia a estudiar Escrig Sos et al. [2021]).

2.1.1. Antecedentes. Un poco de historia

Aunque el concepto metaanálisis surgió por primera vez en 1976 de la mano de Gene Glass, existen ejemplos del uso de algunos de sus fundamentos desde el siglo XIX por parte de Gauss o Laplace entre otros.

Tras estos, Pearson, a principios del siglo XX, realizó una publicación sobre la efectividad de una vacuna contra la fiebre tifoidea en los soldados de la armada británica. En ella combinó diversos estudios pequeños y también se interesó por las causas de la variabilidad entre los resultados de los estudios (heterogeneidad), dos de las cuestiones centrales del metaanálisis. Además, mostró sus resultados en una tabla que podría considerarse la precursora de los diagramas de efecto (*forest plot*) actuales.

Poco después, Ronald Fisher, desarrolló el análisis de la varianza (ANOVA) con el que analizar múltiples estudios relacionados con el efecto de los fertilizantes en la agricultura. Además, animó en publicaciones posteriores a que los investigadores publicasen sus resultados de forma rigurosa y clara, con el fin de poder realizar comparaciones y aplicar estas técnicas de metaanálisis.

Otro hito se dio en los años 40 cuando Joseph Gaither y Joseph Banks realizaron, en el área de psicología, el primer intento de revisión sistemática donde aparecía la estimación de la influencia de los estudios no publicados (sesgo de publicación).

Más adelante, en 1972, el epidemiólogo británico Archie Cochrane demostró que se estaba perdiendo información de interés para la clase médica al no reunir los resultados de los ensayos clínicos aleatorizados. Este fue el punto de inflexión que propició el desarrollo de dos iniciativas que han fomentado la expansión del uso del metaanálisis: *The Cochrane Collaboration* y *The Evidence Based Medicine (EBM)*.

Otra cuestión relevante que se dio durante los años 80 fue el desarrollo y discriminación entre los modelos de efectos fijos (FEM), como el llevado a cabo por Peto, y los modelos de efectos aleatorios (REM) como el realizado por Rebecca DerSimonian y Nan Laird en 1986.

No obstante, hasta la década de 1990 no se extendió la utilización del metaanálisis y hoy en día es una herramienta muy valiosa para la síntesis de evidencia en la investigación médica y de salud. De hecho, se utiliza en estudios de efectividad de tratamientos, evaluación de diagnósticos y muchas otras áreas de investigación (Borenstein et al. [2009]).

Aún hoy en día se siguen desarrollando nuevas técnicas y analizando y mejorando las existentes. Si se requiere de más información acerca de la historia del metaanálisis se recomienda consultar Higgins [2018].

2.1.2. Problemas y objetivos

Una de las principales causas del cambio de paradigma que llevó a la implantación definitiva del uso del metaanálisis a finales del siglo XX fue la infoxicación, esto es, la imposibilidad de procesar y analizar las enormes cantidades de información ofrecidas por los estudios que se estaban llevando a cabo anualmente. No en vano, desde 1980 hasta nuestros días, se produce un crecimiento exponencial del número de estudios publicados.

En muchas ocasiones son tantos los estudios sobre la misma temática que pueden llegar a ofrecer resultados opuestos, lo que hace difícil discernir (sin realizar una revisión) cuál de ellos es un resultado válido para toda la población (si es que alguno lo es).

Por otra parte, las revisiones narrativas realizadas por expertos suelen pecar de ser demasiado subjetivas en la selección de los artículos así como difícilmente reproducibles, introduciendo un importante sesgo en los resultados finales ofrecidos (Escrig Sos et al. [2021]).

Así, el metaanálisis ofrece una salida para estos problemas siendo estos los principales objetivos que se marca:

- **Mejora de la potencia y la precisión**. En ocasiones hay estudios que poseen un tamaño muestral pequeño o no suficientemente grande como para poder detectar diferencias estadísticamente significativas entre las características o grupos que se desean comparar. Al realizar un metaanálisis se están aglutinando los resultados de varios

estudios al mismo tiempo lo que permite aumentar ese tamaño muestral y disponer de una mayor confianza y precisión en los resultados y conclusiones ofrecidos.

- **Exploración de la heterogeneidad**. Es bastante frecuente encontrar estudios que ofrecen resultados diversos acerca de una misma cuestión, llegando en ocasiones a parecer incluso inconsistentes entre sí. La medición de la heterogeneidad evalúa cómo de diferentes son los resultados y, en caso de ser muy diferentes se deberán explicar las causas de dichas diferencias.

- **Análisis y reducción de sesgos**. El último objetivo es doble. Por un lado, consiste en la eliminación de los posibles sesgos existentes en la selección de los artículos escogidos, recogiendo todos los que cumplen unos criterios prefijados claros y objetivos. Por otro lado, hay que realizar un análisis del sesgo de publicación, lo que significa que hay que comprobar si los estudios del metaanálisis podrían tender hacia un determinado resultado debido a que, en muchas ocasiones, solo se publican los estudios que ofrecen resultados significativos en detrimento de los que no.

2.1.3. Tipos de metaanálisis según el tipo de resultado del estudio

El metaanálisis se desarrolló principalmente para el estudio de resultados epidemiológicos por lo que, aunque su uso se haya expandido a otras áreas, muchas de sus características y de las explicaciones de su funcionamiento están basadas en conceptos estadísticos relacionados con este ámbito. Es por esta razón por lo que generalmente se habla de su uso en estudios experimentales (es decir, ensayos epidemiológicos aleatorizados) y en estudios observacionales (por ejemplo, los de prevalencia de una enfermedad). En los estudios experimentales suele aplicarse un metaanálisis tradicional, mientras que en los observacionales es más común la utilización de un metaanálisis de proporciones. No obstante esto no puede considerarse como una regla general, ya que hay ocasiones en las que pueden ser intercambiables.

Aunque poseen muchas características comunes, se deben tener en cuenta ciertos detalles que diferencian a ambos metaanálisis. El metaanálisis tradicional pretende analizar el efecto de una variable independiente en un resultado o efecto final (por ejemplo si la aplicación de una vacuna ha provocado una menor tasa de mortalidad), de modo que se puede evaluar claramente unos valores como positivos y otros como negativos. Mientras que en el caso de las proporciones la idea es estimar una proporción general de un resultado específico (cantidad de personas enfermas sobre el total de la población, por ejemplo).

A lo largo de los siguientes apartados se irán puntualizando estas diferencias, algunas de las cuales habrá que tenerlas muy en cuenta a la hora de obtener las conclusiones del metaanálisis.

2.1.4. Medidas de efecto

Se suele definir una **medida de efecto** (también conocida como tamaño del efecto) a la relación entre dos variables de una población o muestra. Aquí cabe hacer un inciso indicando que algunos autores solo consideran medidas de efecto aquellas que se refieren a los resultados en los que se aplica algún tipo de intervención o tratamiento a los participantes del estudio y su comparación con un grupo de control. En este trabajo se tomará una definición más amplia del concepto, del mismo modo que se hace en Harrer et al. [2022], donde se considerará cualquier efecto sobre una variable sin necesidad de realizar ninguna intervención. Incluso se va a usar el término para referirnos a la medidas de tendencia central (medias o proporciones) que obviamente hacen referencia a una única variable.

Estas medidas de efecto van a depender de dos cuestiones: el tipo de variable respuesta (binaria/dicotómica o continua) y del diseño del estudio.

Así, en el caso del metaanálisis tradicional, las medidas que generalmente se toman son los índices de riesgo absolutos y relativos, que suelen dividirse entre los que están basados en la diferencia de ambas variables (diferencia de riesgo, diferencia de medias, ...) y los que se basan en el cociente o proporción de las mismas (riesgo relativo, *odds ratio*, ...).

Cuando se trata con respuesta dicotómica las medidas de efecto más comunes suelen ser la diferencia de riesgo, el riesgo relativo y el *odds ratio*. Mientras que cuando aparece la respuesta continua, las medidas de efecto más ampliamente usadas son la diferencia de medias y la diferencia de medias estandarizada [1].

En la Tabla 2.1 se muestra el tipo de tabla que suele usarse en el metaanálisis tradicional.

Tabla 2.1: Tabla de contingencia 2x2

	Grupo Tratamiento	**Grupo Control**
Evento	a	b
No Evento	c	d

Y algunas de las medidas anteriormente nombradas se calculan como sigue:

- **Diferencia de riesgos**. Se define como la diferencia de la probabilidad de sufrir un evento entre los elementos del grupo de tratamiento y la probabilidad de sufrir un evento entre los elementos del grupo de control:

$$DR = \frac{a}{a+c} - \frac{b}{b+d}$$

- **Riesgo relativo**. Se trata del cociente entre la probabilidad de sufrir un evento por parte de los individuos que están en el grupo de tratamiento y la probabilidad de sufrir un

[1] https://www.sergas.es/Saude-publica/Documents/1930/11-Ayuda Meta-anlisis.pdf

evento de entre los individuos que están en el grupo de control:

$$RR = \frac{\frac{a}{a+c}}{\frac{b}{b+d}} = \frac{a(b+d)}{b(a+c)}$$

- ***odds ratio***. Se puede usar tanto en estudios prospectivos como retrospectivos o transversales. En el caso retrospectivo, se define como el cociente de *odds* de estar en el grupo de tratamiento en los individuos con evento y el *odds* de estar en el grupo de tratamiento en los individuos sin evento; mientras que en el prospectivo se define como el cociente de *odds* de tener un evento en los individuos del grupo de tratamiento y el *odds* de tener un evento en los individuos del grupo de control. Se puede comprobar que son medidas equivalentes y su método de cálculo sería el siguiente:

$$OR = \frac{ad}{bc}$$

Hay ocasiones donde es necesario aplicar una transformación logarítmica, como ocurre en el caso de las dos últimas medidas, para poder transformarla en una variable aleatoria con distribución normal, ya que por defecto son asimétricas (rango $(0, \infty)$).

Para el caso de metaanálisis de proporciones, se van a utilizar medidas de efecto que toman la proporción de una determinada magnitud (casos) sobre el total de la población. Posibles ejemplos son la prevalencia, la especificidad o la sensibilidad. En estos casos también se puede utilizar una tabla de contingencia 2x2 (ver Tabla 2.2) para realizar los cálculos, normalmente conocida como matriz de confusión. En esta tabla se puede observar que los valores considerados como reales son Enfermo/No enfermo y los valores predichos son Detectado/No detectado. Además, VP representa al verdadero positivo (se ha detectado correctamente un enfermo), VN al verdadero negativo (se ha detectado correctamente una persona sana), FP se refiere al falso positivo (se ha detectado como enfermo una persona que no lo está realmente) y FN al falso negativo (se ha detectado como sana a una persona que en realidad está enferma).

Tabla 2.2: Matriz de confusión. Verdadero Positivo (VP), Falso Positivo (FP), Falso Negativo (FN), Verdadero Negativo (VN)

	Enfermo	**No enfermo**
Detectado	VP	FP
No detectado	FN	VN

Así se pueden calcular algunas de estas medidas de proporción del siguiente modo:

- **Prevalencia**. Mide la proporción de individuos que presentan una determinada carac-

terística sobre el total. Usando la tabla anterior, estar enfermos:

$$Prevalencia = \frac{VP + FN}{VP + FN + VN + FP}$$

- **Sensibilidad**. Calcula el porcentaje de casos positivos detectados como enfermos.

$$Sensibilidad = \frac{VP}{VP + FN}$$

- **Especificidad**. Calcula el porcentaje de casos negativos no detectados como enfermos.

$$Especificidad = \frac{VN}{VN + FP}$$

De este modo, cuando se realiza un metaanálisis lo que se pretende es **calcular una medida de efecto global o conjunta** que contempla la aportación de cada una de las medidas de efecto individuales de cada estudio ponderadas mediante unos determinados pesos, los cuales se van a calcular en función de los modelos y transformaciones utilizados. Así, la potencia de la medida de efecto global proporciona una mayor precisión que si se toman las medidas de estudios aislados. Por contra, hay que tener en cuenta que no hay que quedarse únicamente con el resultado de la medida de efecto global, ya que en dicho caso se podría estar perdiendo información relevante de ciertos estudios o subgrupos de estudios que podrían analizarse o convendría tener en cuenta. Se debe intentar tener una visión completa del metaanálisis y ofrecer unas conclusiones en las que también se valoren dichas cuestiones.

2.1.5. Modelo de efectos

Una decisión que hay que tomar a la hora de realizar un metaanálisis es el tipo de modelo de efectos a utilizar. Se distinguen principalmente dos modelos clásicos a usar en el metaanálisis: el modelo de efectos fijos (FEM) y el modelo de efectos aleatorios (REM). No obstante, también existen otros dos que pueden ser útiles en determinadas circunstancias: el modelo de efectos mixto (MEM) y el modelo bayesiano que no se tratarán aquí.

El modelo de efectos fijos (FEM)

Según Borenstein et al. [2009], este modelo asume que hay un único tamaño de efecto real que comparten todos los estudios del metaanálisis, de modo que todas las diferencias observadas en los tamaños de efecto de dichos estudios respecto a este tamaño de efecto real (o equivalentemente la varianza de los tamaños de efectos observados) son debidas al error de muestreo inherentes a los propio estudios. Así, el modelo se describiría del siguiente modo:

$$Y_i = \theta + \epsilon_i$$

donde

- Y_i es el efecto observado del estudio i-ésimo del metaanálisis.
- θ es el tamaño de efecto real.
- ϵ_i es el error de muestreo del estudio i-ésimo con $\epsilon_i \sim N(0, \sigma^2_{\epsilon_i})$.

Existen diversos métodos que implementan este modelo. Los principales, según Cochrane (Cochrane [2023]), y que están implementados en su aplicación RevMan[2] son:

- **Inverso de la Varianza**. Este es el enfoque más comúnmente utilizado. Consiste en la ponderación de los estudios que forman en metaanálisis a través de la inversa de sus respectivas varianzas. De este modo, los estudios más precisos, es decir, aquellos con menores error estándar, son los que más peso tienen en la estimación global del efecto. Con esta estrategia se consigue minimizar la incertidumbre de la estimación del tamaño del efecto conjunto.
- **Mantel-Haenszel**. Este método es útil cuando se tienen pocos datos o los estudios tienen tamaños pequeños o muy diferentes. El cálculo de la ponderación usada depende de la medida de efecto a utilizar.
- **Peto**. Su uso está restringido a los *odds ratio* y también es apropiado para la combinación de estudios con tamaños muestrales diferentes. Su enfoque es parecido al del Inverso de la Varianza pero aplicando ponderaciones diferentes.

En la Tabla 2.3 puede verse un resumen de qué métodos de efectos fijos se pueden utilizar en función del tipo de dato y la medida de efecto que se esté tratando.

Tabla 2.3: Resumen de los métodos de metaanálisis de efectos fijos disponibles en RevMan. Fuente: (Cochrane [2023])

Tipo de dato	Medida de efecto	Método de efectos fijos
Dicotómico	*odds ratio* (OR)	Mantel-Haenszel
		Peto
		Inverso de la Varianza
	Razón de Riesgo (RR)	Mantel-Haenszel
		Inverso de la Varianza
	Diferencia de riesgos (DR)	Mantel-Haenszel
		Inverso de la Varianza
Continuo	Diferencia de medias (DM)	Inverso de la Varianza
	Diferencia de medias estandarizada (DME)	Inverso de la Varianza

Como se ha comentado anteriormente, el método del Inverso de la Varianza es el usado más frecuentemente como modelo de efectos fijos a la hora de realizar un metaanálisis, es

[2]https://training.cochrane.org/online-learning/core-software/revman

por ello que va a ser el que va a ser explicado a continuación.

Dado que se desconoce el tamaño de efecto real, se deberá partir de los estudios del metaanálisis y realizar una estimación del mismo a través del tamaño del efecto conjunto denotado por $\bar{Y}$ y calculado así:

$$\bar{Y} = \frac{\sum_{i=1}^{k} w_i Y_i}{\sum_{i=1}^{k} w_i}$$

donde cada w_i representa el peso con el que se pondera el estudio i-ésimo del metaanálisis en la estimación conjunta y se calcula como la inversa de la varianza de dicho estudio, es decir:

$$w_i = \frac{1}{V[Y_i]}$$

Así, la varianza del tamaño de efecto conjunto será:

$$V[\bar{Y}] = \frac{1}{\sum_{i=1}^{k} w_i}$$

Y su intervalo de confianza al 95 % vendrá dado por:

$$\left(\bar{Y} - 1{,}96 \frac{1}{\sqrt{\sum_{i=1}^{k} w_i}}, \bar{Y} + 1{,}96 \frac{1}{\sqrt{\sum_{i=1}^{k} w_i}} \right)$$

El modelo de efectos aleatorios (REM)

Siguiendo con Borenstein et al. [2009], el modelo de efectos aleatorios supone que los estudios dentro de un metaanálisis tienen suficiente en común como para ser incluidos en él, pero que no puede asumirse la idea de que el tamaño del efecto real sea exactamente el mismo en todos los estudios. Esto ocurre en el caso de que haya algún tipo de característica que provoque esta diferencia entre estudios (por ejemplo que el rango de edad o el estado de salud sea diferente de un estudio a otro), es decir, que las poblaciones usadas en los distintos estudios difieren. Así existirá una variabilidad real entre los tamaños de efecto observados entre los estudios (heterogeneidad) y deberá ser incluida en el modelo, que se describiría como:

$$Y_i^* = \theta + \epsilon_i + \zeta_i$$

donde

- Y_i^* es el efecto observado del estudio i-ésimo del metaanálisis.
- θ es el tamaño de efecto real.
- ϵ_i es el error de muestreo del estudio i-ésimo con $\epsilon_i \sim N(0, \sigma_{\epsilon_i}^2)$.

- ζ_i es la variación del estudio i-ésimo con respecto al tamaño de efecto real y donde $\zeta_i \sim N(0, \tau^2)$.

Ahora se realiza una estimación del tamaño de efecto real a través del tamaño del efecto conjunto denotado por $\bar{Y^*}$ y calculado así:

$$\bar{Y^*} = \frac{\sum_{i=1}^{k} w_i^* Y_i^*}{\sum_{i=1}^{k} w_i^*}$$

donde cada $w*_i$ representa el peso con el que se pondera el estudio i-ésimo del metaanálisis en la estimación conjunta, y se calcula como la inversa de la suma la varianza de dicho estudio y la estimación de la varianza de la heterogeneidad, es decir:

$$w_i^* = \frac{1}{V^*[Y_i]} = \frac{1}{V[Y_i] + \hat{\tau}^2}$$

Así, la varianza del tamaño de efecto conjunto será:

$$V[\bar{Y}] = \frac{1}{\sum_{i=1}^{k} w_i^*}$$

Y su intervalo de confianza al 95 % vendrá dado por:

$$\left(\bar{Y} - 1{,}96 \frac{1}{\sqrt{\sum_{i=1}^{k} w_i^*}}, \bar{Y} + 1{,}96 \frac{1}{\sqrt{\sum_{i=1}^{k} w_i^*}} \right)$$

Al igual que ocurría en el caso del modelo de efectos fijos, existen diferentes métodos que implementan este modelo. Su principal diferencia consiste en la forma de estimar la varianza de la heterogeneidad (τ^2). Los más comunes son los siguientes:

- **DerSimonian-Laird (DL)**. Es, con diferencia, el más usado y el utilizado por defecto. Cuando se habla de modelo de efectos aleatorios sin especificar el método, se están refiriendo al algoritmo DerSimonian-Laird. Está basado en el método del Inverso de la Varianza, pero realiza un ajuste en la ponderación para incluir el grado de heterogeneidad. Así, cuando en el metaanálisis no existe heterogeneidad, este esquema es equivalente al Inverso de la Varianza. Por otro lado, si aparece heterogeneidad, los intervalos de confianza del tamaño de efecto conjunto serán mayores que al usar el modelo de efectos fijos estándar. Para estimar τ^2 se usa la siguiente fórmula:

$$\hat{\tau}^2 = max \left\{ 0, \frac{Q - (k-1)}{\sum_{i=1}^{k} w_i - \frac{\sum_{i=1}^{k} w_i^2}{\sum_{i=1}^{k} w_i}} \right\}$$

donde Q es el estadístico de contraste de la homogeneidad siguiente:

$$Q = \sum_{i=1}^{k} w_i (Y_i - \bar{Y})^2$$

- ***Maximum Likelihood* (ML)** o ***Restricted Maximum Likelihood* (REML)**. En estos casos se utilizan los enfoques de máxima verosimilitud y máxima verosimilitud restringida para obtener la varianza asociada a la heterogeneidad. Una vez calculada dicha varianza el proceso a seguir es equivalente al usado en el método DerSimonian-Laird.

- **Paule-Manel (PM)**. Tal y como se indica en Evangelou and Veroniki [2022], este método está basado en una versión generalizada del estadístico Q que da lugar a la siguiente expresión:

$$\sum_{i=1}^{k} w_i^* (Y_i - \bar{Y}^*)^2 = k - 1$$

o equivalentemente:

$$\sum_{i=1}^{k} \frac{(Y_i - \bar{Y}^*)^2}{V[Y_i] + \hat{\tau}^2} = k - 1$$

El valor de $\hat{\tau}^2$ se obtiene mediante un proceso iterativo de la fórmula anterior hasta que el método converja, tomando cero como valor inicial de $\hat{\tau}^2$.

Consejos para la elección del modelo

La primera idea que debe quedar clara es que la elección del modelo debe realizarse siempre a priori, no se puede realizar un estudio y escoger posteriormente una de las dos medidas en función de los resultados obtenidos. Escoger el modelo evaluando los resultados ofrecidos por un test de heterogeneidad no es correcto.

Se puede optar por el modelo de efectos fijos siempre y cuando se considere que se han utilizado todos los estudios disponibles y que estos son equivalentes, es decir, que se han llevado a cabo bajo las mismas condiciones. Una segunda cuestión a tener en cuenta para escoger este tipo de modelo es que el tamaño del efecto conjunto obtenido se podrá aplicar únicamente a poblaciones idénticas a las de los estudios, nunca se podrá generalizar a otras poblaciones.

Por contra, cuando se conozca que no se dispone de todos los estudios o que parte de los mismos no son equivalentes, la utilización del modelo de efectos aleatorios será más adecuada. Además en este caso sí que se podrá generalizar y extrapolar la medida calculada a otras poblaciones.

En general, la recomendación va a ser utilizar el modelo de efectos aleatorios, ya que suele ser bastante complicado que se cumplan las condiciones del modelo de efectos fijos. Sin

embargo, si el número de estudios a utilizar es muy pequeño puede haber un problema ya que, en ese caso, la estimación de la varianza entre los estudios (τ^2) tendrá una precisión muy pobre. En Borenstein et al. [2009] se proponen tres posibles soluciones a esta situación, aunque ninguna de ellas es perfecta:

- Mostrar las medidas de efecto separadas pero no calcular la medida de efecto conjunta.
- Cambiar a un modelo de efectos fijos. Siempre teniendo en cuenta que no se podrá inferir a otras poblaciones el resultado.
- Realizar una aproximación bayesiana, estimando el valor de τ^2 a través de datos externos al conjunto de estudios usados en el metaanálisis. Esta es la opción preferida por los autores aunque conlleva el problema de conocer como llevar a cabo un metaanálisis bayesiano.

2.1.6. Diagrama de efectos o *forest plot*

La principal herramienta gráfica que permite evaluar los efectos individuales y conjuntos de los estudios de un simple vistazo es el denominado *forest plot*.

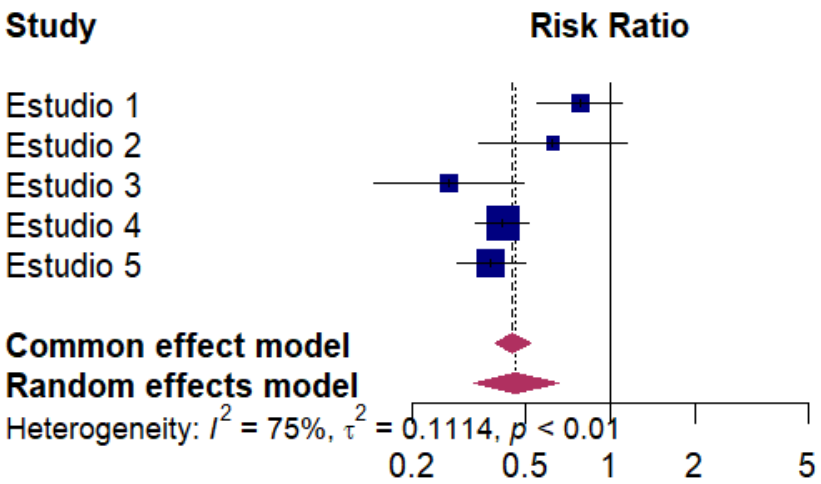

Figura 2.2: Ejemplo de diagrama *forest plot*

A continuación se va a explicar, en base a la Figura 2.2, como interpretar este diagrama.

Lo primero que cabe destacar es que el metaanálisis del ejemplo está formado por 5 estudios nombrados como «Estudio 1», ..., «Estudio 5». En cada una de las filas que representa al estudio se observa un cuadrado de color azul que es el estimador puntual del efecto (el riesgo relativo en este caso) junto con unas líneas horizontales o bigotes que representan el intervalo de confianza alrededor de esta estimación. Lo habitual, si no se indica lo contrario, es utilizar

un intervalo de confianza del 95 %. Otra cuestión a destacar es el tamaño del cuadrado, los de mayor tamaño serán aquellos que disponen de un tamaño muestral superior.

La siguiente información que proporciona el diagrama son los diamantes morados. Estos diamantes representan el intervalo de confianza de los tamaños de los efectos conjuntos, siendo el superior el resultado obtenido usando el modelo de efectos fijos y el inferior el alcanzado usando el modelo de efectos aleatorios. Se observan también dos líneas verticales que parte del centro de cada uno de los rombos, estas serían las estimaciones puntuales conjuntas.

Se puede ver que aparecen también datos que hacen referencia a la heterogeneidad, por ejemplo el valor de τ^2. En el próximo apartado se explicará con detalle como analizar la heterogeneidad observando el diagrama y los valores que nos ofrece.

2.1.7. Análisis de la heterogeneidad

La heterogeneidad mide el grado de variabilidad existente entre los resultados de los distintos estudios. En apartados anteriores se han avanzado algunas de las medidas a utilizar para comprobar la heterogeneidad. Aquí, se profundizará en algunas de ellas y se dará su interpretación.

Los diagramas *forest plot* que se muestran a continuación van a permitir distinguir, de un vistazo, el tipo de heterogeneidad presente en el metaanálisis.

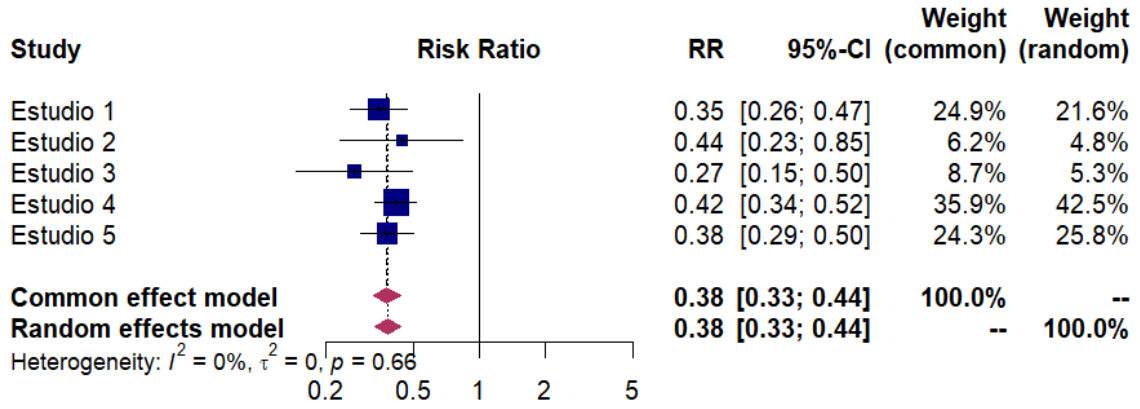

Figura 2.3: Ejemplo de diagrama *forest plot* homogéneo

Para poder distinguir, mediante un diagrama de efectos, si el metaanálisis es homogéneo o heterogéneo, hay que prestar atención a si la estimación del modelo de efecto conjunto (la línea vertical del diamante del modelo de efecto que se esté usando) se cruza con los intervalos de confianza de las medidas de efecto de cada uno de los estudios (los bigotes de cada caja).

Así, en la Figura 2.3 se observa que, en efecto, todos los intervalos se cortan con la estimación conjunta, por lo que el metaanálisis es claramente homogéneo. Otra idea a destacar es

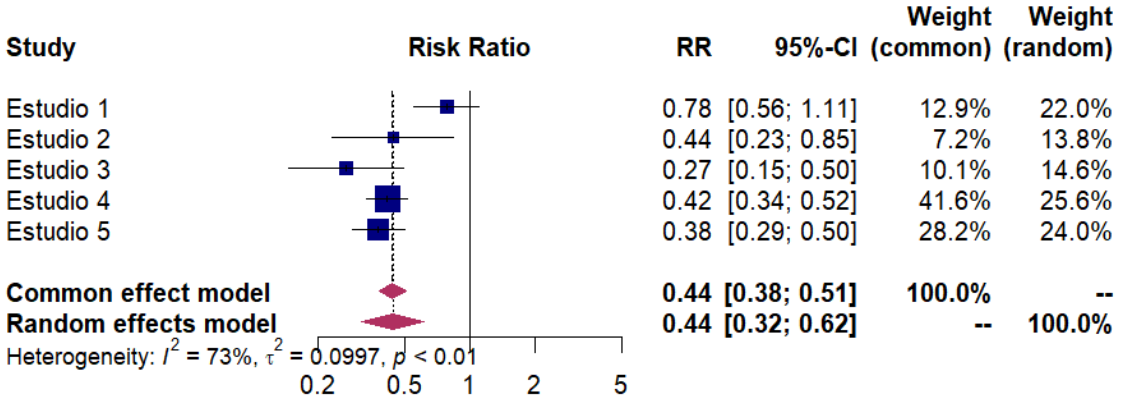

Figura 2.4: Ejemplo de diagrama *forest plot* heterogéneo concordante

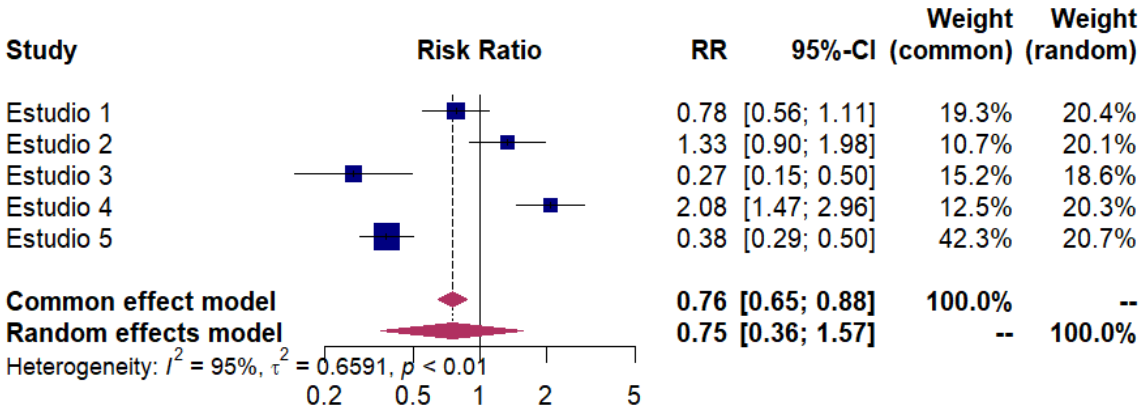

Figura 2.5: Ejemplo de diagrama *forest plot* heterogéneo discordante

que los diamantes de ambos modelos se encuentran en la misma posición y tienen la misma forma y tamaño, ya que, cuando hay homogeneidad ambos modelos coinciden.

Por otro lado, en la Figura 2.4 y la Figura 2.5 se puede comprobar que existe algún estudio que no corta la estimación conjunta, por lo que en ambos casos se trata de metaanálisis con heterogeneidad.

Ahora bien, dentro de los metaanálisis tradicionales (no así en los metaanálisis de proporción) se pueden distinguir dos tipos de heterogeneidad: la concordante (o clínica) y la discordante (o metodológica). En el primer caso se trata de una heterogeneidad que tiene que ver con alguna característica del tipo de elementos de los estudios (como podría ser la edad o el estado de salud de los participantes). Este tipo de heterogeneidad es tratable en el sentido de que se pueden dar las razones de su aparición y tomar las medidas adecuadas. Esta situación se puede observar en el diagrama de efectos si todos los estudios se encuentran al mismo lado respecto de la línea vertical que aparece en $RR = 1$.

Por otro lado, la heterogeneidad discordante es la más difícilmente resoluble pues existen estudios que están a favor y otros que están en contra (a ambos lados de la barra verti-

cal de $RR = 1$) y puede deberse a razones muy variadas que van desde los diseños de los estudios hasta las poblaciones utilizadas. Con este tipo de metaanálisis no se suele poder llegar a una conclusión efectiva sin realizar la aplicación de técnicas más avanzadas como la estratificación, el análisis de subgrupos o la metarregresión.

Medidas de cuantificación de la heterogeneidad

Cuando se realiza un metaanálisis se espera que los estudios sean homogéneos entre sí, por lo que se buscará que la heterogeneidad sea la menor posible. Hay que tener en cuenta que la estimación conjunta ofrecida por el metaanálisis solo será válida si el grupo de estudios que lo forman es suficientemente homogéneo. Para cuantificar la heterogeneidad se dispone de una serie de test y medidas con las que reconocer el grado de heterogeneidad existente.

La primera medida para valorar la heterogeneidad, que ya se ha adelantado en la Sección 2.1.5, es la estimación de la varianza del tamaño de efecto real, denominada $\hat{\tau}^2$. Si esta medida toma el valor cero, el grupo de estudios del metaanálisis es homogéneo, y cuanto más lejano sea a este valor, mayor será la heterogeneidad existente. Por desgracia, observando esta medida de manera independiente no se puede clasificar el grado de heterogeneidad existente ya que depende de la escala en la que se miden los efectos.

Otra forma de valorar la existencia de heterogeneidad es el estadístico de contraste Q, que como ya se definió en la Sección 2.1.5 tiene la siguiente fórmula de cálculo:

$$Q = \sum_{i=1}^{k} w_i (Y_i - \bar{Y})^2$$

Se trata de un estadístico con una distribución χ^2 con $k - 1$ grados de libertad (donde k es el número de estudios que forman el metaanálisis) bajo la hipótesis nula de homogeneidad. Así, si este test ofrece un p-valor bajo habría indicios de heterogeneidad. Se ha de tener en cuenta que este test posee una potencia estadística baja cuando el número de estudios o el tamaño de la muestra que utilizan estos es pequeño, por ello, en lugar del clásico valor de 0,05 se suele tomar un valor de 0,1 como significativo a partir del cual no se puede rechazar la hipótesis nula.

Por último, existe una medida independiente de la escala que permite cuantificar la cantidad de heterogeneidad existente, denominada I^2. Esta medida representa la proporción de la varianza del tamaño del efecto conjunto que puede ser debida a la heterogeneidad. Su cálculo se realiza del siguiente modo:

$$I^2 = \frac{Q - (k-1)}{Q} 100$$

Por su naturaleza, la interpretación de I^2 es más sencilla que la de τ^2. Así, según Cochrane [2023], una guía aproximada para medir la heterogeneidad será la siguiente:

- $0\,\% - 40\,\%$: Heterogeneidad no relevante
- $30\,\% - 60\,\%$: Heterogeneidad moderada
- $50\,\% - 90\,\%$: Heterogeneidad significativa
- $75\,\% - 100\,\%$: Heterogeneidad considerable

La elección de uno u otro intervalo va a depender del peso que tengan los test de heterogeneidad (por ejemplo el p-valor dentro del test Q). Por ello, es muy aconsejable calcular las tres medidas que se han desarrollado en este apartado para hacer un análisis de la heterogeneidad con mayor exactitud.

Hay que hacer un apunte aquí, según Barker et al. [2021] no existen test específicos para la comprobación de la heterogeneidad en los metaanálisis de proporciones, por lo que se sugiere la utilización del estadístico I^2. Sin embargo, por la forma en la que está construido y debido a la naturaleza de los datos proporcionales que suelen ofrecer una varianza pequeña, el valor de I^2 es alto normalmente para este tipo de metaanálisis. Además, el artículo explica que es esperable que exista cierta heterogeneidad a la hora de calcular las medidas de efecto de los metaanálisis proporcionales y que valores altos de I^2 no necesariamente deben implicar que haya inconsistencia entre los estudios, así que se debe de analizar este resultado con cierta cautela.

Tratamiento de la heterogeneidad

Una vez que se ha comprobado la existencia de heterogeneidad existen diversas formas de actuar para tratarla:

- **Explorar las causas**. Una de las cuestiones que conviene realizar es revisar las características de los distintos estudios con el fin de intentar encontrar las posibles causas que permitan explicar la heterogeneidad. Tras ello, se pueden realizar dos técnicas principalmente: realizar una metarregresión o realizar un análisis de subgrupos. Esta última técnica consiste en realizar un metaanálisis por subgrupos de estudios o estratificado tomando en cuenta las características que han podido causar la heterogeneidad.
- **Excluir estudios**. La heterogeneidad puede ser debida a la presencia de varios estudios atípicos. La exclusión de estos estudios debe realizarse con razones de peso para no introducir sesgo en el análisis.
- **No usar los resultados del metaanálisis**. Si a pesar de todo la heterogeneidad sigue siendo considerable, habría que tomar la decisión de no considerar el resultado final obtenido en el metaanálisis y quedarnos simplemente con la revisión sistemática de los estudios, incluyendo también las posibles causas de la heterogeneidad.

2.1.8. Análisis del sesgo de publicación

El sesgo de publicación se puede producir si no se tienen en cuenta los estudios que no se han considerado en el metaanálisis debido a que no han sido publicados. Por fortuna, existen formas de estimar el número aproximado de estudios no publicados. Además, para identificar la existencia de sesgo de publicación se dispone del conocido como gráfico en embudo o *funnel plot* (ver Figura 2.6), que presenta la dispersión entre los resultados de cada estudio y el error estándar o alguna medida relacionada.

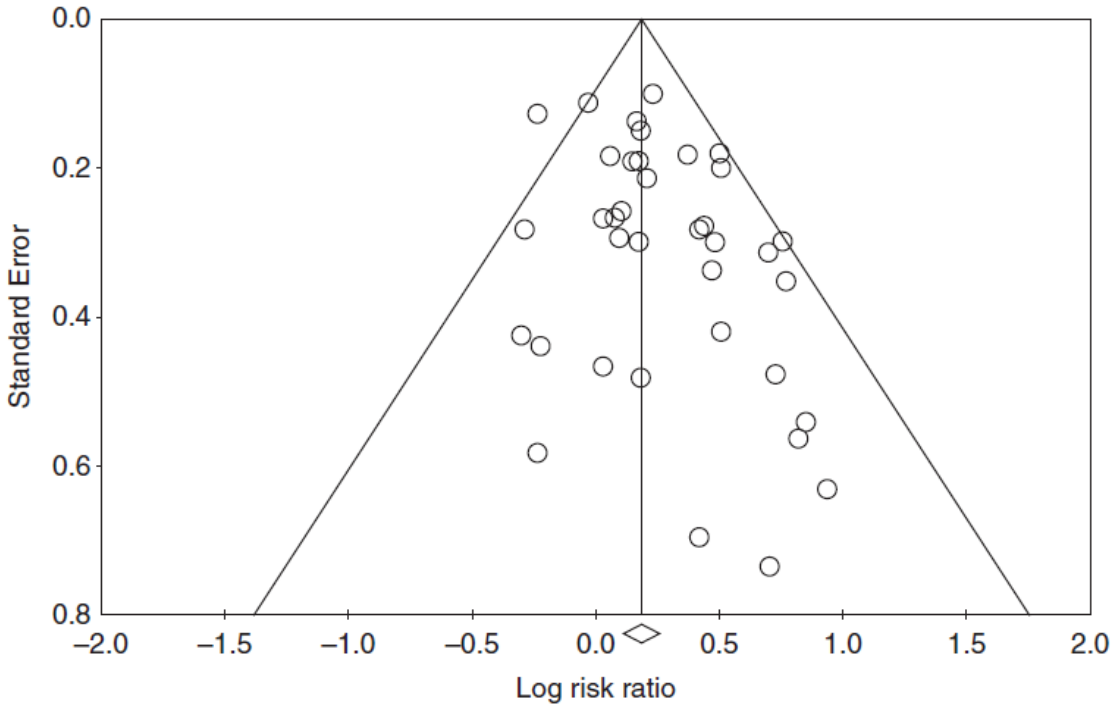

Figura 2.6: Ejemplo de diagrama de embudo. (Fuente: Borenstein et al. [2009])

Como se observa, en el eje horizontal se coloca la medida del efecto que se contrapone al error estándar en el eje vertical. Cada punto representa un estudio dentro del metaanálisis. Los estudios más grandes suelen estar arriba al poseer menor varianza y los más pequeños abajo.

Se puede afirmar que existe sesgo si la figura que conforman los puntos muestran cierta asimetría. En el caso de la imagen del ejemplo se puede comprobar que en los estudios de la parte baja tienen cierta tendencia a aparecer más en la parte derecha, por lo que se atisba algo de sesgo cuando los estudios son pequeños.

Una cuestión a tener en cuenta con los *funnel plot* es la necesidad de que existan más de 10 estudios, ya que en caso contrario será difícil comprobar la existencia de asimetrías.

Aparte del diagrama de embudo, también se puede hacer uso de algunos estadísticos que permiten evaluar el sesgo como son: la prueba de Begg y la prueba de Egger. La prueba de Begg estudia la presencia de correlación entre las estimaciones de los tamaños de efectos y

sus varianzas. De modo que si el test afirma que hay correlación entonces probablemente exista sesgo. Por otro lado, el test de Egger analiza, mediante una regresión lineal, si hay relación entre los tamaños de efecto y el error estándar. En ambos casos habrá que considerar un p-valor significativo de 0,10 y la potencia estadística que presentan ambos es baja cuando hay pocos estudios.

Según Barker et al. [2021], los test de Egger y Begg y el diagrama *funnel plot* fueron desarrollados para usarse con datos comparativos ya que asumen la mayor frecuencia de estudios con resultados favorables que estudios negativos. Por ello, dado que en un metaanálisis de proporciones no existe un consenso de qué se puede considerar un resultado positivo, desaconseja realizar este tipo de análisis con estas herramientas y en su lugar llevar a cabo una evaluación cualitativa.

2.1.9. Metaanálisis de proporciones

A la hora de llevar a cabo un metaanálisis de proporciones simple habrá que tener en cuenta que se deberá escoger entre dos tipos de modelos principales: el Inverso de la Varianza (que ya se explicó en la Sección 2.1.5) y el Modelo Lineal Mixto Generalizado (GLMM).

En el caso del Inverso de la Varianza, dado que las proporciones $p = \frac{a}{n}$, donde a es el número de eventos del estudio y n es el tamaño de la muestra del estudio, ofrecen valores en el intervalo $[0, 1]$, es muy conveniente realizar una transformación previa a los datos para evitar problemas que pueden surgir con el intervalo de confianza fuera de ese rango. Así, las transformaciones más comunes son las siguientes:

- **Logit**. Se utiliza frecuentemente también en el metaanálisis tradicional (por ejemplo en el riesgo relativo o en el *odds ratio*). Su cálculo se realiza como sigue:

$$\theta_{logit} = ln\left(\frac{p}{1-p}\right)$$

  De modo que su rango de valores pasa a ser $(-\infty, \infty)$. Su varianza sería:

$$V[\theta_{logit}] = \frac{1}{np} + \frac{1}{n(1-p)} = \frac{1}{np(1-p)}$$

  Por lo que en los casos en los que el número de eventos ($a = np$) sea cero, la varianza tomaría el valor infinito. Para evitar estas situaciones se suele sumar 0,5 tanto al numerador como al denominador de la fracción.

- **Arcoseno**. Esta transformación, también clásica, se calcula como sigue:

$$\theta_{arcsen} = arcsen(\sqrt{p})$$

Su rango de valores varía entre 0 y $\pi/2$. La varianza de esta transformada será:

$$V[\theta_{arcsen}] = \frac{1}{4n}$$

Por tanto, no necesita de una estabilización como sí necesitaba la transformación logit.

- **Freeman-Tukey (FT) o doble arcoseno**. Se trata de una variante de la anterior que se define así:

$$\theta_{FT} = 0{,}5\left(arcsen\left(\sqrt{\frac{a}{n+1}}\right) + arcsen\left(\sqrt{\frac{a+1}{n+1}}\right)\right)$$

Y su varianza es:

$$V[\theta_{arcsen}] = \frac{1}{4n+2}$$

Esta prueba ha demostrado (Evangelou and Veroniki [2022]) tener mejores propiedades de estabilización de la varianza para tamaños de muestra pequeño que la anterior.

Se da por hecho que todas ellas se aproximan a una distribución normal, por lo que su intervalo de confianza al 95 % tendrá la forma:

$$(\theta - 1{,}96EE[\theta], \theta + 1{,}96EE[\theta])$$

En el caso de que las muestras sean pequeñas se deberá usar una distribución binomial en su lugar.

Una vez realizados los cálculos propios del metaanálisis se debe llevar acabo una transformación inversa de cada una de estas transformaciones para volver a obtener los datos en el intervalo $[0, 1]$.

- **Inversa de logit**.

$$p_{logit} = \frac{1}{1 + e^{-\theta_{logit}}}$$

- **Inversa de arcoseno**.

$$p_{arcsen} = sen(\theta_{arcsen})^2$$

- **Inversa de Freeman-Tukey**. Su formulación es mucho más compleja que las anteriores:

$$p_{FT} = 0{,}5\left\{1 - signo(cos(2\theta_{FT}))\sqrt{1 - \left(sen(2\theta_{FT}) + \frac{sen(2\theta_{FT}) - \frac{1}{sen(2\theta_{FT})}}{n}\right)^2}\right\}$$

En cuanto al **Modelo Lineal Mixto Generalizado (GLMM)**, se trata de una extensión del Modelo Lineal Generalizado (GLM) donde se incluyen los tamaños de efecto aleatorios además de los fijos. En el ámbito del metaanálisis de proporciones, se fundamenta en el supuesto de que el número de eventos (a) de cada estudio sigue una distribución binomial. Así, GLMM hace uso de un modelo de regresión logística (se recomienda consultar Gareth et al. [2013] para obtener información sobre este modelo) para ajustar los datos.

2.2. *Malware*

El término *malware* proviene de la contracción en inglés de las palabras *malicious software*, que en español se puede traducir, según la Fundéu [3], como programa maligno o malicioso. Como se indica en Mell et al. [2005] un *malware* es un «programa informático que es introducido en un sistema, normalmente de forma encubierta, con la intención de comprometer la confidencialidad, la integridad o la disponibilidad de los datos, aplicaciones o sistema operativo de la víctima, o de molestarla o perturbarla de algún modo». Existen diversas categorías de *malware*, entre las principales están:

- **Virus**. Permanecen latentes en el sistema hasta que alguna acción humana (normalmente la ejecución de un fichero infectado) los activan. Una vez activados, están diseñados para autorreplicarse y distribuir sus copias a otros archivos, programas o sistemas.
- **Gusanos**. Al igual que los virus, son capaces de autorreplicarse y distribuir sus copias, sin embargo, no es necesaria ninguna actuación humana para desencadenar estas acciones.
- **Troyanos**. Son programas que tienen apariencia de benignos pero que realizan acciones maliciosas de forma oculta. Al contrario que virus y gusanos no son capaces de autorreplicarse.
- ***Adware***. Está diseñado para mostrar anuncios automáticamente en la interfaz del usuario.
- ***Spyware***. Normalmente viene oculto junto con otros programas instalados. Su objetivo es recopilar información con el fin de enviarla a terceros.
- ***Ransomware***. Este tipo de *malware* secuestra los archivos (cifrándolos, por ejemplo) o los accesos al sistema para después forzar al usuario a pagar un rescate por ellos.
- ***Rootkit***. Consiste en una colección de ficheros instalados en un sistema para alterar su funcionalidad y adquirir permisos de administrador, de modo que le permite realizar cambios o instalar aplicaciones.

[3] https://www.fundeu.es/recomendacion/programa-maligno-mejor-malware/

- ***Criptojacking***. Permite el acceso a los recursos de su sistema para llevar a cabo minería de criptomonedas.

Los ataques *malware* pueden estar dirigidos a cualquier tipo de dispositivo: desde ordenadores personales a servidores, pasando por dispositivos móviles y cualquier otro tipo de dispositivo electrónico. Aunque se puede desarrollar *malware* que ataque a hardware, lo más común es que se realice a través de algún tipo de plataforma o sistema operativo. De este modo, dada su enorme implantación mundial, Windows es, con bastante diferencia sobre el resto, la plataforma que más ataques de este tipo tiene registrados, mientras que otros sistemas como MacOs o Linux no sufren tanto este tipo de ataques. Por otro lado, la expansión de sistemas operativos de uso específico como Android ha favorecido el desarrollo de nuevo *malware* orientado a dispositivos móviles, colocándose en segunda posición respecto al número de ataques recibidos, aunque aún están muy lejos de las cifras de Windows (AV-Test Institute [2023]).

2.2.1. Formas de detección

Análisis de malware

«El análisis del *malware* es el proceso de determinar el propósito y las características de un *malware*» (Nirav Bhojani [2014]). Durante este proceso se trata de descubrir ciertas cuestiones como podrían ser: el tipo de *malware* y su capacidad de infección, su estructura interna, así como el tipo de sistema que puede ser atacado. Nirav Bhojani [2014], define dos categorías de análisis: la estática y la dinámica. Sin embargo, en la actualidad (Tahir [2018], Sihwail et al. [2018], Sehrawat and Singh [2022]) hay un mayor consenso por incluir una tercera categoría: la híbrida. En muchas ocasiones también se menciona una cuarta opción (el análisis de memoria) pero podría ser considerado como una variante del análisis dinámico por lo que no va a ser tratado aquí.

- **Análisis Estático**. Este tipo de análisis es el que comprueba cómo de malicioso es el software a través de sus características sin llegar a ejecutarlo, simplemente examinando su código. Sus principales ventajas son: es seguro, al no tener que ejecutar el software; permite desvelar estructura, patrones y comportamiento del *malware* a través de su código y flujo de ejecución; y es más rápido que el análisis dinámico. Por otra parte, sus principales desventajas son: poca efectividad para detectar *malware* ofuscado o que utiliza otras técnicas de evasión, y escasa capacidad para capturar comportamientos que ocurren en tiempo de ejecución lo que puede dar lugar a falsos negativos y a desconocer el impacto sobre el sistema.

- **Análisis Dinámico**. Este análisis implica ejecutar el software en un entorno controlado (generalmente una máquina virtual tipo *sandbox*) para así observar su compor-

tamiento durante su ejecución. Entre sus ventajas se encuentran: mejor rendimiento ante *malware* desconocido o con técnicas evasivas, y un conocimiento del uso que hace del sistema ya que observa su comportamiento. Algunas de sus desventajas son: mayor riesgo al tener que ejecutarlo, mayor lentitud en comparación con el análisis estático, y que puede tener problemas con *malware* que reconoce y bloquea entornos de análisis.

- **Análisis Híbrido**. Combina aspectos del análisis estático y dinámico con el fin de obtener una visión más completa del posible *malware*. Entre sus ventajas se encuentran que mejora la precisión reduciendo tanto falsos positivos como falsos negativos; y que puede analizar *malware* polimórfico, ofuscado y otras técnicas de evasión. Sus desventajas son: necesidad de mayores recursos y tiempo, así como mayor complejidad de uso que puede requerir la atención de expertos.

En la Tabla 2.4 se resumen las ventajas y desventajas que poseen cada uno de los tipos de análisis anteriormente descritos.

Tabla 2.4: Comparación entre los distintos tipos de análisis

Tipo de análisis	Ventajas	Desventajas
Análisis Estático	Rapidez	Ofuscación y técnicas de evasión
	Estructura y patrones	Falsos negativos
	Seguridad	Impacto sobre el sistema
Análisis Dinámico	*Malware* desconocido	Mayor Riesgo
	Técnicas de evasión	Lentitud
	Comportamiento	Bloqueo de entornos de análisis
Análisis Híbrido	Mejor precisión	Lentitud
	Malware polimórfico	Mayor uso de recursos
	Malware ofuscado	Mayor complejidad

Técnicas de detección

A continuación se muestran las principales técnicas utilizadas por cada uno de los tipos de análisis. Para el caso del análisis estático se tienen las siguientes:

- **Análisis de firmas**. La firma del software no es más que un conjunto de características del fichero como pueden ser su tamaño, las funciones importadas o exportadas o la posición de determinados bytes. Este análisis consiste en extraer la firma y comprobar si se encuentra en un determinado repositorio o base de datos que contiene las firmas de los *malware* registrados hasta el momento.

- **Extracción de cadenas**. Se trata de identificar cadenas de texto concretas que pueden contener comandos para realizar funciones maliciosas o bien incluir URLs, nombres de ficheros o direcciones IP de lugares sospechosos.

- **Desensamblado y descompilado**. Consiste en traducir el lenguaje máquina del software a lenguaje ensamblador en el primero de los casos, o al código fuente en caso de que el software estuviera escrito en un lenguaje de alto nivel. De este modo se pueden estudiar las instrucciones y la lógica del programa.
- **Análisis de flujo**. Mediante esta técnica consistente en la creación de grafos de control de flujo de ejecución, los analistas pueden estudiar la estructura del *malware* y conocer mejor su posible comportamiento.

Cuando se utiliza el análisis dinámico las principales técnicas son las siguientes:

- **Monitorización**. Se observa la actividad del sistema (uso de red, memoria, CPU, ...) así como posibles modificaciones de archivos y registros.
- **Depuración**. Se examina el flujo del programa durante su ejecución con el fin de detectar instrucciones que provoquen un comportamiento extraño.
- **Análisis de memoria**. Se revisa el contenido de la memoria con el fin de descubrir instrucciones potencialmente sospechosas que puedan suponer una amenaza. Mediante esta técnica es posible la detección de *malware* ofuscado.

Para llevar a cabo las distintas técnicas se usan herramientas específicas como: desensambladores y descompiladores, depuradores de código, máquinas virtuales tipo *sandbox*, herramientas de análisis de red y herramientas que analizan el comportamiento y antivirus.

2.2.2. Ataques que se escapan a la detección

Uno de los mayores desafíos a los que se enfrentan los expertos en ciberseguridad es la capacidad de los ataques de *malware* para escapar a la detección. A continuación, se explorarán las principales técnicas que utilizan los ciberdelicuentes para evadir la detección, así como posibles técnicas desarrolladas para evitar estos ataques.

Encriptación y ofuscación de malware

Se trata de dos de las principales técnicas utilizadas para ocultar o dificultar el análisis de código malicioso.

La **encriptación** consiste en la utilización de algoritmos de cifrado como AES o RSA para proteger el código parcial o totalmente. Su principal objetivo es evitar su reconocimiento mediante técnicas de análisis estático como el análisis de firmas o la extracción de posibles patrones. Suelen infectar los sistemas tras desencriptarse a sí mismos usando el algoritmo de desencriptado y la clave. Una vez realizada la infección, generan una nueva clave con la que volver a encriptarse y replicarse o expandirse a otros lugares sin ser detectado (Sahay et al. [2020]).

La **ofuscación** realiza una modificación del *malware*, de modo que su estructura sea más compleja y confusa, para evitar que las herramientas como los antivirus sean capaces de detectarlos a través de su firma o que los investigadores sean capaces de comprender su funcionalidad y objetivos (Andrade et al. [2022]). Entre las posibles técnicas de ofuscación están: el renombrado de variables y funciones, la división del código malicioso en fragmentos más pequeños y su dispersión, la introducción de código basura o la alteración del flujo de ejecución añadiendo saltos y bucles (Zhang et al. [2021a]).

Entre las posibles contramedidas para evitar ataques que usen encriptación y ofuscación destacarían: la realización de un análisis dinámico para poder detectar acciones sospechosas, la aplicación de técnicas de aprendizaje automático que puedan encontrar patrones y comportamientos maliciosos, y el mantenimiento de software actualizado con la aplicación de los parches necesarios.

Malware oligomórfico, polimórfico y metamórfico

Son variantes de programas maliciosos que son capaces de modificarse dinámicamente para evitar su detección. Usan técnicas que permiten cambiar su apariencia y comportamiento con cada infección, lo que dificulta su detección a través de las firmas que usan los antivirus.

El ***malware*** **oligomórfico** es aquel que genera versiones ligeramente diferentes de sí mismo en cada versión aunque sigue manteniendo la misma funcionalidad básica. La ventaja de este tipo de *malware* respecto a la técnica de encriptación es que incluyen el algoritmo de cifrado/descifrado y son capaces de modificarlo en cada instancia para hacer más compleja su detección. Sin embargo, su capacidad de cambio está limitada a unos cientos de modificaciones. (Sahay et al. [2020]).

El ***malware*** **polimórfico** a diferencia del anterior modifica algunas partes adicionales de su código y su estructura sin afectar a su comportamiento. Además, es capaz de generar millones de modificaciones no solo cientos.

El ***malware*** **metamórfico** es el más avanzado de los tres ya que es capaz de reescribir completamente su código. Esta transformación provoca cambios no solo en su estructura sino también en su lógica y comportamiento.

Las posibles medidas a aplicar para tratar a este tipo de *malware* son muy similares a las usadas para evitar las técnicas de ofuscación y encriptación: aplicación de un análisis dinámico, utilización de firmas heurísticas que sean capaces de detectar patrones y comportamientos maliciosos en tiempo de ejecución, aplicación de técnicas de aprendizaje automático y mantenimiento de software actualizado y aplicación de parches.

Ataques de día cero

Los ataques de día cero son aquellos que se lanzan sobre una vulnerabilidad de los sistemas que no ha sido detectada por los desarrolladores y, por tanto, no dispone de un parche que

la corrija. Este tipo de ataques no deja rastro y puede permanecer oculto durante años hasta su descubrimiento (Sibi Chakkaravarthy et al. [2019]).

Dada su naturaleza se trata de ataques muy difíciles de prevenir. Las técnicas usadas para evitarlos comprenden: la aplicación de parches y actualizaciones rápidas una vez se ha descubierto; los análisis de comportamiento y la aplicación de técnicas de aprendizaje automático para tratar de identificar actividades sospechosas; así como la evaluación de riesgos y la aplicación de medidas proactivas para mejorar el nivel de protección.

Ataques adversarios

Se trata de una forma avanzada de evasión y se refiere al conjunto de tácticas usadas por los atacantes para engañar a las soluciones de seguridad. Los ataques adversarios más comunes en este contexto son los referidos a los modelos de aprendizaje automático, llamados ataques adversarios en aprendizaje automático.

Los ataques adversarios en aprendizaje automático consisten en la manipulación de los datos que se les suministra a la red (ya sea en su fase de entrenamiento o en fases posteriores), mediante aplicación de ruido u otras técnicas, de modo que provocan la clasificación del dato suministrado de forma errónea. Se suele decir que dichos datos generan una ilusión a la red neuronal de modo similar a lo que nos ocurre a los humanos con las ilusiones sensoriales (ópticas, auditivas, ...).

Según Liu et al. [2022], los ataques adversarios se pueden clasificar en función de distintos parámetros:

- **Según su objetivo**. Si la intención del atacante es hacer que el modelo prediga un valor erróneo independientemente de cual sea, entonces se está ante un **ataque no dirigido**. En cambio, si el atacante precisa que la respuesta del modelo ante cierta entrada sea una en particular (por ejemplo, si quiere que un clasificador de imágenes de razas de perro catalogue erróneamente a los pastores alemanes), entonces se trata de un **ataque dirigido**.

- **Según el tipo de información que posee el atacante**. En el caso de que el atacante desconozca cualquier información del modelo incluida su arquitectura o el conjunto de datos de entrenamiento, se denominaría **ataque de caja negra**. En este caso solo podrá manipular muestras de entrada y observar la salida que ofrece el modelo para intentar encontrar algún patrón que le permita generar muestras adversarias. Por el contrario, sería un **ataque de caja blanca**, si el atacante conoce el modelo, sus parámetros y características de entrada. Aquí el atacante sería capaz de generar muestras falsas fácilmente que pueden engañar al modelo.

- **Según el propósito del ataque**. El **ataque de envenenamiento**, consiste en la inclusión de muestras adversarias envenenadas (es decir, etiquetadas de forma errónea

o bien añadiéndoles algún tipo de perturbación) durante la fase de entrenamiento de modo que se reduzca la precisión del modelo. El **ataque de evasión** consiste en la inclusión de ruido en la entrada del modelo para que clasifique de forma incorrecta y se suele realizar tras el entrenamiento para extraer información del modelo.

Existen diversas formas de defenderse de los ataques adversarios, en Shaukat et al. [2022] destacan las siguientes: entrenamiento con muestras adversarias, enmascaramiento del gradiente, destilación defensiva y reducción de características del modelo.

2.2.3. Técnicas de Aprendizaje Automático para la detección

En Técnicas de detección se ha realizado una clasificación de técnicas basada en el tipo de análisis. Sin embargo, es posible llevar a cabo otro tipo de clasificación tal y como se explica en Sahay et al. [2020]:

- **Detección basada en firmas**. Es la aproximación tradicional que se corresponde con el análisis de firmas estático que se explicó anteriormente.
- **Detección basada en heurísticos**. Se distinguen dos vertientes. Por un lado, los métodos de análisis estáticos que analizan los programas con el fin de obtener ciertos patrones que permitan identificarlos como *malware*. Por otro, los métodos dinámicos que ejecutan el software en un entorno controlado para observar su comportamiento. En este grupo también aparece el análisis y aplicación de reglas de detección basadas en conocimiento experto.
- **Normalización de *malware***. Esta técnica consiste en la normalización de los programas a través de la eliminación de la ofuscación que tenga aplicada el código, así como de la detección de variaciones entre programas. Incluso permite detectar variantes desconocidas de *malware*.
- **Aplicación de técnicas de aprendizaje automático**. Se realizan normalmente dos pasos: el primero consiste en la extracción de características del *malware* que se tomarán del conjunto de datos que se use para entrenar el modelo; y el segundo que consiste en entrenar al modelo que se escoja (árboles de decisión, redes neuronales, SVM, redes neuronales profundas, ...) con dichos datos. A través de estas técnicas se consigue detectar nuevas variantes de *malware* desconocidas.

2.2.4. Modelos de Aprendizaje Profundo

El aprendizaje profundo (*deep learning*) es una rama del aprendizaje automático que utiliza redes neuronales con múltiples capas ocultas (generalmente más de tres) con el fin de extraer patrones complejos de los datos. Sus múltiples capas permiten ampliar en gran

número la cantidad de neuronas y conexiones existentes entre ellas respecto a una red neuronal tradicional. Esto facilita que tengan una mayor capacidad de digerir grandes volúmenes de datos y, a partir de ellos, realizar un aprendizaje de tareas complejas con alto nivel de abstracción.

Existen multitud de modelos de aprendizaje profundo, algunos están especializados en tareas muy específicas como son el procesamiento y generación de imágenes, el procesamiento del lenguaje natural, la traducción o el reconocimiento de voz. Sin embargo, aquí se van a mostrar solo algunos de ellos, los más utilizados para la detección de *malware*. Cabe destacar que, en bastantes ocasiones, estos modelos se pueden unir para atacar la detección de *malware* de distintas maneras al mismo tiempo. No obstante, en este trabajo solo se van a presentar algunas nociones de los modelos básicos, no se tratarán sus posibles combinaciones.

Red Neuronal Convolucional (CNN)

Su diseño está inspirado en el estudio del córtex visual del cerebro y es ampliamente utilizada para el procesamiento de imágenes y tareas relacionadas con la visión por ordenador. No obstante se puede usar para cualquier tipo de datos que tenga topología de malla. Su arquitectura está compuesta de varias capas:

- **Capas de convolución**. Son el elemento básico de esta red. Esta capa es la que se encarga de extraer las características más relevantes a través de una serie de filtros (llamados *kernels*). Normalmente va seguida de una capa de activación que le permite mejorar sus capacidades.

- **Capas de agrupamiento**. Se encargan de reducir la dimensionalidad proveniente de los mapas de características devueltos por las capas de convolución con el fin de optimizar la eficiencia de la red.

- **Capa completamente conectada**. Tras la concatenación de varias capas de convolución y agrupamiento, se agrega una capa totalmente conectada que realiza la clasificación o la detección.

Durante el proceso de entrenamiento las capas iniciales de convolución aprenden las características más básicas y conforme se va avanzando a través de capas se van obteniendo características cada vez más complejas y completas.

Dentro de la detección de *malware* este tipo de redes suele usarse en la técnica del análisis por imagen. Consiste en tratar a los ficheros binarios como imágenes (codificando cada byte como un píxel en una imagen) y aplicar las CNN para extraer características y patrones complejos que permitan la detección (Yadav and Tokekar [2021]).

Red Neuronal Recurrente (RNN)

Es un tipo de red que se enfoca en el manejo de datos secuenciales como pueden ser las series temporales o el propio lenguaje natural. Así, es muy usado en el reconocimiento de voz o los traductores.

Su arquitectura está basada en células de memoria que no son más que neuronas que calculan su salida en función del valor actual de la secuencia de entrada en el paso t y el valor de la salida obtenida por la célula en el paso $t-1$. Su característica diferenciadora es que poseen retroalimentación, lo que les permite tener esa especie de memoria, mientras que su mayor limitación viene dado por el problema del desvanecimiento y explosión del gradiente que se da cuando los gradientes de error se tornan muy pequeños o muy grandes a medida que se propagan.

Una de las variantes principales de las RNN que es capaz de resolver este problema es la red de memoria a corto plazo de larga duración (*Long Short Term Memory*: LSTM). Su arquitectura se basa tres capas: una de entrada, una oculta completamente conectada y otra de salida. En la capa oculta se encuentran las células de memoria, estas disponen de mecanismos que permiten controlar cuando debe memorizarse nueva información, cuando puede borrarse y cuando puede ser usada como salida.

Autoencoder (AE)

Se trata de una técnica de aprendizaje no supervisado. Su objetivo consiste en extraer las características más importantes de los datos. Para ello realiza un proceso de codificación de las entradas y posterior descodificación que intenta reconstruir los datos originales. Así, los *autoencoders* constan de dos capas en su arquitectura:

- Capa de codificación (*encoder*). Se encarga de tomar los datos de entrada y mapearlos hacia una representación de características comprimida.
- Capa de descodificación (*decoder*). Reconstruye los datos originales a través de la descompresión de la representación comprimida generada por la capa de codificación.

Además, existe una función de pérdida que evalúa la diferencia existente entre los datos originales y los reconstruidos y es la que permite ajustar los pesos a la red.

Para una revisión en mayor profundidad de estas y otras redes de aprendizaje profundo se recomiendan las siguientes lecturas: Weidman [2019], Géron [2019] y Goodfellow et al. [2016].

Capítulo 3

Materiales y Métodos

El método central que se ha seguido para dar respuesta a los objetivos marcados en este trabajo consiste en la realización de una revisión sistemática con metaanálisis. Para ello se ha utilizado PRISMA como guía fundamental. PRISMA es acrónimo de *Preferred Reporting Items for Systematic Reviews and Meta-Analyses* (Elementos de Informe Preferidos para Revisiones Sistemáticas y Metaanálisis) y se marca como principal objetivo mejorar la transparencia y la calidad de los informes de revisiones sistemáticas y metaanálisis.

PRISMA proporciona una lista detallada de 27 elementos que los autores pueden utilizar para asegurarse de que su informe de metaanálisis cumpla con los estándares recomendados. La lista de comprobación incluye elementos que deben añadirse en el informe, como el diseño de la revisión, los criterios de selección de estudios, los métodos de búsqueda, los métodos de evaluación de calidad, los métodos de síntesis de evidencia, los resultados del análisis y las conclusiones. Además, se incluyen 7 cuadros que ofrecen una explicación más detallada de ciertos aspectos como son la metodología y el proceso que se debe seguir durante la revisión sistemática (Urrútia and Bonfill [2010]).

Siguiendo las indicaciones de PRISMA, durante las fases de identificación y selección de artículos es conveniente seguir el diagrama de flujo de información que se facilita en la Figura 3.1. Puede encontrarse mucha más información en su página web [1]

Por otra parte, y como se comentará más adelante, se han utilizado diversos motores de búsqueda de artículos científicos para localizar los estudios concernientes a la aplicación de algoritmos de aprendizaje profundo en la detección de *malware* y, de ellos, se han clasificado y seleccionado los más relevantes.

Asimismo, para la realización de los diversos experimentos efectuados y la obtención de sus resultados gráficos y numéricos se ha utilizado la herramienta de programación estadística R.

[1] http://www.prisma-statement.org/

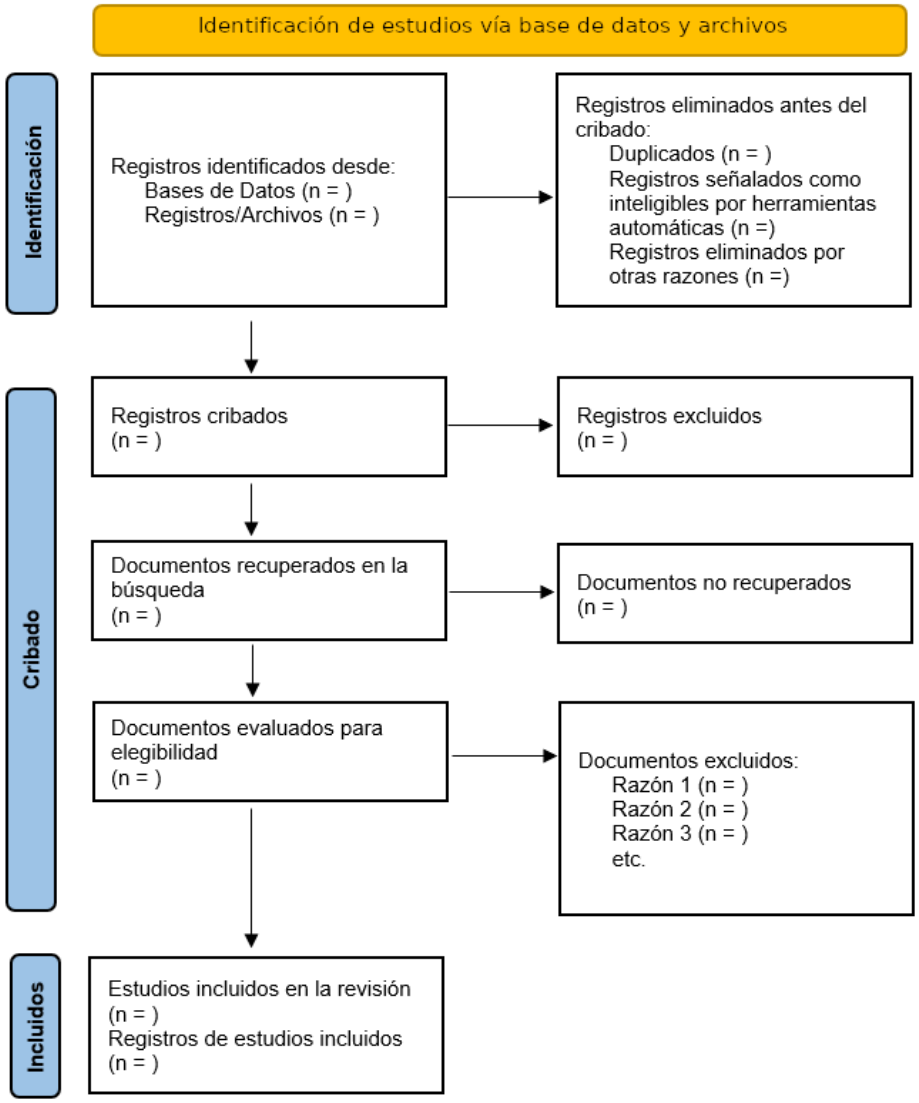

Figura 3.1: Diagrama de flujo de información PRISMA

3.1. Realización de un metaanálisis paso a paso

A modo de resumen, se enumeran los pasos que deben realizarse a la hora de llevar a cabo cualquier metaanálisis y que van a ir siendo desarrollados a lo largo de este capítulo y los siguientes. Cabe destacar que muchos de los elementos aquí indicados van a ser comunes a la realización de una revisión sistemática, ya que todo metaanálisis parte de una revisión sistemática previa. Por otro lado, aunque no se haga referencia explícitamente en este trabajo, dentro de estos pasos se van a ir aplicando los criterios de la lista PRISMA.

1. **Preparación**. En esta primera etapa se deberán definir una serie de cuestiones clave en todo proceso de metaanálisis. Lo primero será decidir cuál es la pregunta que se va a intentar resolver. Además, hay que realizar un diseño previo del metaanálisis definiendo cuestiones como: la estrategia de búsqueda, los criterios de inclusión, exclusión

y elegibilidad para los estudios, la información a extraer de los estudios y cómo se van a presentar los resultados. La primera fase será la realización de una pregunta.

2. **Identificación y búsqueda de estudios**. Durante esta fase se debe realizar una búsqueda sistemática en diversas bases de datos científicas y almacenar los resultados obtenidos para su procesamiento en la siguiente fase.

3. **Selección de estudios**. A partir de los resultados anteriores se debe de realizar un filtrado de los estudios en función de los criterios de inclusión, exclusión y elegibilidad que se han definido en el paso de preparación.

4. **Extracción de datos**. Se obtiene y almacena adecuadamente la información relevante procedente de los estudios que han sido elegidos para el metaanálisis.

5. **Análisis estadístico**. A continuación se realiza el análisis estadístico que consiste en la evaluación de la heterogeneidad, la evaluación de los sesgos y, si es posible, el cálculo de la estimación de la medida de efecto conjunta.

6. **Conclusiones**. Finalmente, se interpretan y discuten los resultados obtenidos, y se ofrecen unas conclusiones.

3.1.1. Fase de preparación

Aunque podría considerarse como un paso previo a esta fase, la primera dificultad encontrada a la hora de realizar este trabajo fue la elección del tema. Fue una tarea ardua que llevó a analizar diversas alternativas antes de escoger finalmente la detección de *malware* como punto central.

Una idea fundamental de la que se partía es que se pretendía realizar un metaanálisis sobre un tema que utilizase técnicas de aprendizaje profundo. Para ello, se revisaron temas relacionados con la salud (nutrición, ejercicio, consumo de medicamentos y suplementos, enfermedades varias), con la educación, con la economía, con la naturaleza y el cambio climático, con las fuentes renovables, con los medios de comunicación y las *fake news*, entre otras cuestiones para comprobar que ya existían metaanálisis al respecto en la mayor parte de casos o que el número de estudios era insuficiente como para llevarlo a cabo. Finalmente, el ámbito de la ciberseguridad parecía disponer de suficientes estudios como para realizar un metaanálisis.

Dentro de un ámbito tan amplio como es la ciberseguridad, hubo que realizar diversas consultas para ceñirse a un tema en particular, ya que el número de resultados obtenidos era inabarcable en el tiempo que se disponía. Ahí fue donde se decidió focalizar los esfuerzos únicamente en la detección de *malware* mediante el uso de herramientas de aprendizaje profundo.

El objetivo principal estaba definido a partir de ese momento, se quería comprobar si la aplicación de técnicas de aprendizaje profundo permitían obtener buenos resultados en la detección de *malware* en función de los estudios realizados hasta entonces.

3.1.2. Fase de identificación y búsqueda de estudios. Fuentes de información

Para llevar a cabo esta fase, los motores de búsqueda utilizados fueron Scopus, PubMed e IEEE Xplore, así como Google Académico como base de datos de apoyo. Las búsquedas se basaron en tres grandes conceptos: *malware*, detección y redes de aprendizaje profundo. Así, se utilizarán como palabras clave *malware*, los distintos tipos de redes de aprendizaje profundo junto con las siglas por las que son conocidas y la palabra detectar y sus derivadas. A la hora de llevar a cabo la búsqueda, no se impuso ningún criterio temporal y se incluyeron todas las publicaciones desde la primera fecha disponible hasta abril de 2023, pero sólo se tomaron en cuenta los artículos, no se contemplaron las conferencias, libros o revistas como resultados válidos. Además, se llevó a cabo una eliminación de los artículos repetidos una vez finalizada la búsqueda bibliográfica.

Se presentan a continuación las consultas que se llevaron a cabo en cada uno de los motores indicados:

Scopus

```
(TITLE-ABS-KEY ("deep learning")
 OR TITLE-ABS-KEY ("Convolutional neural network")
 OR TITLE-ABS-KEY ("Deep belief network")
 OR TITLE-ABS-KEY ("recurrent neural network")
 OR TITLE-ABS-KEY ("deep neural network")
 OR TITLE-ABS-KEY ("generative adversarial network")
 OR TITLE-ABS-KEY ("long short term memory")
 OR TITLE-ABS-KEY ("radial basis function network")
 OR TITLE-ABS-KEY ("multilayer perceptron")
 OR TITLE-ABS-KEY ("self organizing map")
 OR TITLE-ABS-KEY ("restricted boltzmann machine")
 OR TITLE-ABS-KEY ("autoencoders")
 OR TITLE-ABS-KEY ("CNN") OR TITLE-ABS-KEY ("DBN")
 OR TITLE-ABS-KEY ("RNN") OR TITLE-ABS-KEY ("DNN")
 OR TITLE-ABS-KEY ("GAN") OR TITLE-ABS-KEY ("LSTM")
 OR TITLE-ABS-KEY ("RBFN") OR TITLE-ABS-KEY ("MLP")
 OR TITLE-ABS-KEY ("SOM") OR TITLE-ABS-KEY ("RBM"))
AND (TITLE-ABS-KEY ("malware"))
AND (TITLE-ABS-KEY ("detection") OR TITLE-ABS-KEY ("detect"))
```

Pubmed

```
((deep learning[Title/Abstract])
 OR (convolutional neural network[Title/Abstract])
 OR (Deep belief network[Title/Abstract])
 OR (recurrent neural network[Title/Abstract])
 OR (deep neural network[Title/Abstract])
 OR (generative adversarial network[Title/Abstract])
 OR (long short term memory[Title/Abstract])
 OR (radial basis function network[Title/Abstract])
 OR (multilayer perceptron[Title/Abstract])
 OR (self organizing map[Title/Abstract])
 OR (restricted boltzmann machine[Title/Abstract])
 OR (autoencoders[Title/Abstract])
 OR (CNN[Title/Abstract]) OR (DBN[Title/Abstract])
 OR (RNN[Title/Abstract]) OR (DNN[Title/Abstract])
 OR (GAN[Title/Abstract]) OR (LSTM[Title/Abstract])
 OR (RBFN[Title/Abstract]) OR (MLP[Title/Abstract])
 OR (SOM[Title/Abstract]) OR (RBM[Title/Abstract]))
AND (malware[Title/Abstract])
AND ((detection[Title/Abstract]) OR detect[Title/Abstract]))
```

IEEE Xplore

```
(("All Metadata":deep learning)
 OR ("All Metadata":Convolutional neural network)
 OR ("All Metadata":Deep belief network)
 OR ("All Metadata":recurrent neural network)
 OR ("All Metadata":deep neural network)
 OR ("All Metadata":generative adversarial network)
 OR ("All Metadata":long short term memory)
 OR ("All Metadata":radial basis function network)
 OR ("All Metadata":multilayer perceptron)
 OR ("All Metadata":self organizing map)
 OR ("All Metadata":restricted boltzmann machine)
 OR ("All Metadata":autoencoders)
 OR ("All Metadata":CNN) OR ("All Metadata":DBN)
 OR ("All Metadata":RNN) OR ("All Metadata":DNN)
 OR ("All Metadata":GAN) OR ("All Metadata":LSTM)
 OR ("All Metadata":RBFN) OR ("All Metadata":MLP)
 OR ("All Metadata":SOM) OR ("All Metadata":RBM))
AND ("All Metadata":malware)
```

```
19 AND (("All Metadata":detection) OR ("All Metadata":detect))
```

3.1.3. Fase de selección de estudios. Criterios de exclusión

A continuación se enumeran los criterios de exclusión utilizados para eliminar aquellos artículos que no los cumplan:

1. Se han conservado únicamente las publicaciones que estén escritas en inglés o en español.

2. Se han revisado todos los títulos y resúmenes de los artículos para excluir a todas aquellas publicaciones que carecieran de referencias a aprendizaje profundo o detección de *malware*.

3. Se han eliminado todos los artículos pertenecientes a una revista que no poseyera JIF (*Journal Impact Factor*) en el año 2021 (último año del que hay datos disponibles en Clarivate a fecha de escritura de este trabajo: abril de 2023).

4. Todos aquellos resultados relacionados con artículos que no han podido ser obtenidos han sido también excluidos.

5. No se han considerado todos aquellos artículos que fueran estudios comparativos o bien resúmenes de otros artículos.

6. No se han usado aquellos artículos que no traten de *malware* o no usen aprendizaje profundo tras la revisión completa del texto.

7. Se han utilizado únicamente aquellos artículos que solo detecten *malware*. Si también clasifican, se han descartado.

8. Se han conservado aquellas publicaciones que detecten *malware* de forma general, no las que lo hagan únicamente de un tipo de *malware* específico.

9. Se excluyen todos los artículos que tengan en consideración ataques adversarios o ataques de día cero.

10. Solo han sido estimados aquellos artículos que tuvieran explícitamente la matriz de confusión o bien que hubiera podido ser inferida a través de las medidas aportadas.

Destacar que este último criterio fue tomado dado que, aunque se pretendía trabajar con la especificidad y la sensibilidad como medidas, estaba abierta la posibilidad de trabajar con alguna medida adicional, además de que esto permitía asegurar la detección de posibles erratas en los datos mostrados por los artículos. No obstante, tras la revisión de las publicaciones se pudo comprobar que el porcentaje de artículos que contenían dichas medidas de

forma explícita y no la matriz de confusión (o la posibilidad de calcularla) fue prácticamente cero.

Esta etapa es la que más tiempo requirió de todas ya que, pese a tener definidos de forma exhaustiva los criterios de exclusión previamente, la obtención de una gran cantidad de artículos en la fase de elegibilidad, obligó a realizar una revisión muy extensa.

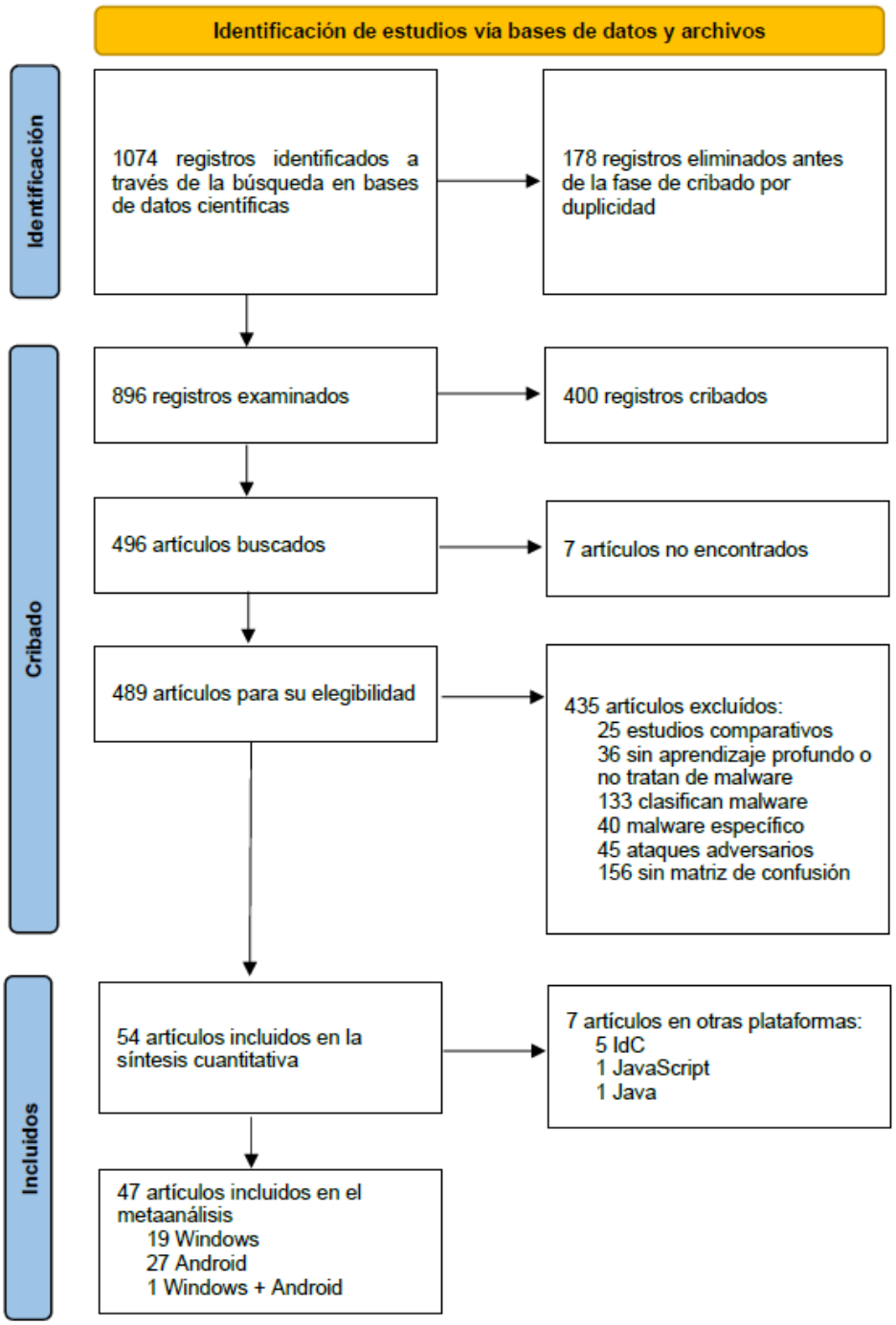

Figura 3.2: Diagrama de flujo de información PRISMA de estudios incluidos

En la Figura 3.2 se muestra el diagrama de flujo con los resultados obtenidos en las distintas fases de la búsqueda para la revisión sistemática y el metaanálisis tal y como se indica en la guía PRISMA. La revisión bibliográfica ofreció 1074 registros de los cuales 861 fueron obtenidos de Scopus, 47 de PubMed y 166 de IEEE Xplore. Tras la eliminación de los 178 registros repetidos quedaron 896 registros para la fase de cribado. De estos, 24 fueron eliminados por no respetar el criterio de idioma (inglés o español), 266 fueron excluidos tras analizar los títulos y resúmenes que no hicieran referencia a técnicas de aprendizaje profundo o a detección de *malware*, y otros 110 fueron descartados por no disponer de JIF en el año 2021. Así, quedaron 496 artículos de los cuales 7 no pudieron ser encontrados, por lo que finalmente se obtuvieron 489 publicaciones elegibles. De ellas, aplicando el quinto criterio, 25 fueron excluidas por tratarse de estudios comparativos; por el sexto criterio, 36 no pertenecían al tema a tratar al no usar técnicas de aprendizaje profundo o bien no tratar de *malware*; 133 se suprimieron debido al séptimo criterio, ya que realizaban clasificación de *malware*; según el octavo criterio, tuvieron que descartarse 40 al centrarse en algún o algunos tipos específicos de *malware* en lugar de tratarlos de forma general; el noveno criterio provocó la eliminación de 45 artículos que trataban de ataques adversarios; por último, el décimo criterio supuso el apartar otras 156 publicaciones que carecían de matriz de confusión o datos suficientes como para calcularla. De modo que, en total se excluyeron 435 artículos quedando únicamente 54 de ellos para ser incluidos en la síntesis cuantitativa. Finalmente, se realiza el metaanálisis con 47 de estas 54 publicaciones tras quitar 7 artículos correspondientes a elementos multiplataforma o que no la indicaban (5 referentes a IdC, uno a JavaScript y otro a Java). De esos 47, 19 son específicos de Windows, 27 de Android y 1 realiza un análisis en ambas plataformas.

3.1.4. Fase de extracción de datos

Durante esta fase (y también durante la anterior) se hizo uso de hojas de cálculo para anotar información acerca de los registros y publicaciones obtenidos en la búsqueda y selección. En el caso particular de los 47 artículos que se incluyeron en el metaanálisis, se llevó a cabo una recolección de características de la publicación, como el título, el autor o el año, así como también de la información que podía ser de utilidad para la realización del metaanálisis como son la plataforma, el tipo de análisis, los modelos de red neuronal usada, los datasets y sus tamaños, la matriz de confusión y las dos métricas utilizadas. En la Tabla 3.1 y la Tabla 3.2 se muestra un resumen de la información más relevante de estos estudios.

Finalmente, parte de esa información se almacenó en un fichero csv separado por punto y coma para su uso en la fase de experimentación. Los campos que contienen dicho formato están indicados en la Tabla 3.3

Tabla 3.1: Resumen datos estudios usados en el metaanálisis

Estudio	**Modelo**	**Plataforma**	**Tipo Análisis**	**VP**	**FP**	**FN**	**VN**
Ye et al. [2017]	AE + RBM	Windows	Estático	489	11	7	493
Zhang et al. [2019]	CNN	Windows	Estático	3065	185	133	3108
Ma et al. [2019]	LSTM	Windows	Estático	78	0	7	103
Čeponis and Goranin [2020]	CNN	Windows	Estático	7634	26	39	1697
Jeon and Moon [2020]	CRNN	Windows	Estático	957	43	32	968
Acarturk et al. [2021]	LSTM	Windows	Dinámico	8705965	13816	70625	2628383
Damaševičius et al. [2021]	MLP + CNN	Windows	Estático	502	1	0	539
Jiang and Zhang [2021]	LSTM	Windows	Estático	8402	134	142	10942
Wang et al. [2021]	CNN	Windows	Dinámico	203	0	4	191
Tian et al. [2021]	CNN	Windows	Dinámico	283	15	13	279
Chen et al. [2022]	CNN + BiLSTM	Windows	Híbrido	9841	159	137	9863
Kim et al. [2022]	CNN + RNN	Windows	Híbrido	35225	295	334	30483
Li et al. [2022]	CNN + BiLSTM	Windows	Dinámico	10282	172	409	10836
Tian et al. [2022]	CNN	Windows	Dinámico	162	2	5	462
Andrade et al. [2022]	LSTM	Windows	Estático	1014	116	138	1028
Xuan et al. [2022]	BiLSTM	Windows	Dinámico	13465	525	704	56821
Ravi et al. [2022]	LSTM + DNN + CNN	Windows	Híbrido	288	15	11	162
Alzahrani et al. [2022]	CNN	Windows	Estático	1443	7	10	140
Catalano et al. [2022]	CNN	Windows	Estático	954	170	79	1113
Du et al. [2023]	TextCNN + DNN	Windows	Dinámico	9763	237	236	9764
Yuan et al. [2016]	DBN	Android	Híbrido	861	19	38	842
Saif et al. [2018]	DBN	Android	Híbrido	321	0	4	121
Kim et al. [2019]	DNN	Android	Estático	2581	26	53	3708
Alotaibi [2019]	LSTM	Android	Estático	1686	167	147	41004

Tabla 3.2: Resumen datos estudios usados en el metaanálisis - continuación

Estudio	**Modelo**	**Plataforma**	**Tipo Análisis**	**VP**	**FP**	**FN**	**VN**
Alzaylaee et al. [2020]	MLP	Android	Híbrido	1146	5	22	1940
Ren et al. [2020]	CNN + RNN	Android	Estático	770	30	37	763
Pei et al. [2020]	GCN + RNN	Android	Estático	9919	81	21	9979
Niu et al. [2020]	LSTM	Android	Estático	436	13	13	237
Zhang et al. [2021b]	TextCNN	Android	Estático	494	6	28	472
Millar et al. [2021]	CNN	Android	Estático	554	2	6	550
Arslan [2021]	DNN	Android	Estático	1302	10	19	194
Zhang et al. [2021c]	TCN	Android	Estático	1112	53	53	1084
Namrud et al. [2021]	DNN	Android	Estático	1275	18	7	2406
Nguyen et al. [2021]	DNN	Android	Estático	5446	100	55	5591
Kong et al. [2022]	FCSCNN	Android	Estático	1461	39	19	1481
Almomani et al. [2022]	CNN	Android	Estático	2925	22	45	452
Wang et al. [2022]	CNN	Android	Híbrido	9247	28	93	9134
Alkahtani and Aldhyani [2022]	LSTM	Android	Híbrido	2836	7	19	1649
Yadav et al. [2022]	CNN	Android	Estático	748	45	19	670
Fallah and Bidgoly [2022]	LSTM	Android	Dinámico	256803	77	154	234623
Yumlembam et al. [2022]	GNN + CNN	Android	Estático	3595	51	30	1079
Yumlembam et al. [2022]	GNN + CNN	Android	Estático	1530	150	76	15194
Wu et al. [2022]	BiLSTM + GNN	Android	Híbrido	874	70	6	923
Ravi et al. [2022]	LSTM + DNN + CNN	Android	Híbrido	671	30	17	675
Kabakus [2022]	CNN	Android	Estático	1401	140	147	1190
Ravi and Chaganti [2022]	CNN	Android	Estático	625	5	10	603
Alomari et al. [2023]	DNN + LSTM	Android	Estático	9980	20	20	9980
Alzubi et al. [2023]	LSTM	Android	Estático	291	5	5	299
Albakri et al. [2023]	ARAE	Android	Estático	3882	33	54	2631

Tabla 3.3: Descripción de los campos del fichero csv con los datos para la experimentación

Nombre	Descripción
Anyo	Año de la publicación
Autor	Autor de la publicación
AutorAnyo	Concatenación del autor y año
CasosPositivos	Número de positivos detectados (TP + FN)
CasosNegativos	Número de negativos detectados (TN + FP)
MuestrasTotal	Número de casos totales (TP + FN + TN + FP)
Plataforma	Plataforma en la que se encuentra el *malware* (Windows o Android)
TipoAnalisis	Tipo de análisis usado en la detección (Estático, Dinámico o Híbrido)
TipoAnalisisNum	Código del tipo de análisis (0 - Estático, 1 - Dinámico, 2 - Híbrido)
Sensibilidad	Valor de sensibilidad en tanto por ciento
Especificidad	Valor de especificidad en tanto por ciento
VP	Verdaderos positivos
FN	Falsos negativos
FP	Falsos positivos
VN	Verdaderos negativos

3.1.5. Análisis estadístico. Experimentos realizados

Para realizar el análisis estadístico se efectuaron tres grandes bloques de experimentos que pretendían dar respuesta a los objetivos marcados:

- **Análisis global**. Es el que contempla a todos los estudios.
- **Estratificado por plataforma**. Se evalúan las dos plataformas (Windows y Android) de forma independiente.
- **Estratificado por plataforma y tipo de análisis**. Se realiza una estratificación a dos niveles, ofreciendo cinco estratos: Windows-Estático, Windows-Dinámico, Windows-Híbrido, Android-Estático y Android-Híbrido. La combinación Android-Dinámico no se realiza al solo poseer un estudio.

En cada uno de estos bloques se llevaron a cabo dos metaanálisis de proporciones, uno para evaluar la sensibilidad y otro para evaluar la especificidad, así, en total se tuvieron 16 resultados. Para obtener esos resultados, se han realizado numerosos experimentos, modificando parámetros como el método (GLMM o Inverso de la Varianza) y las transformación usadas (logit, FT o arcoseno), no obstante, en el apartado de resultados solo se va a mostrar la parametrización que mejores resultados ofrezca, o en su defecto la razón por la que ha sido elegida.

Para la realización de los experimentos se ha utilizado como herramienta principal el programa estadístico R. Además se han instalado los paquetes *meta* y *metafor* que poseen funciones especializadas para la realización de metaanálisis, ya sea tradicional o de proporciones, aunque en este caso solo se usarán las relacionadas con este último tipo. Por otro

lado, dado que son múltiples los experimentos que se han a llevado a cabo y que muchos de ellos poseen un código relativamente similar, se ha programado una función que permite, mediante la elección adecuada de parámetros, llevar a cabo todos y cada uno de los experimentos realizados. En el Apéndice A.1 se muestra el código completo de dicha función. que posee siete parámetros:

- **plataforma**. Se trata de una variable numérica que toma los valores -1, 0 o 1, que representan a todos los estudios, únicamente a los de Windows o sólo los de Android respectivamente.
- **tipoAnalisis**. Es también una variable de tipo entero y sus valores van desde el -1 hasta el 2, siendo el -1 el que representa a todos los estudios independientemente del tipo de análisis, el 0 al análisis estático, el 1 al dinámico y el 2 al híbrido.
- **medida**. Es la métrica que va a usarse en el metaanálisis. En este caso sus valores pueden ser 0, que simboliza a la sensibilidad y 1 que hace lo propio con la especificidad.
- **metodo**. Se refiere al método o a la transformación usada. Así, los valores 0, 1 y 2 aluden al método del inverso de la varianza usando la transformaciones logit, Freeman-Tukey y arcoseno, mientras que el valor 3 indica que se va a usar el método GLMM con la transformación logit.
- **ficheroDatos**. Es el nombre del fichero csv que contiene la información de todos los estudios
- **idExperimento**. Identificador único de cada experimento. Es utilizado para distinguir los nombres de los ficheros imagen descargados tras la ejecución
- **influyentesAEliminar**. Consiste en una lista, que puede ser vacía, con las posiciones de los estudios que van a ser suprimidos del metaanálisis en esa ejecución.

Durante su funcionamiento, este procedimiento realiza cuatro grandes pasos, que permitirán obtener toda la información generada durante un metaanálisis de proporciones:

- **Cálculo de las medidas de efecto**. Aplica el método escogido para obtener tanto las medidas de efecto individuales de los estudios como la medida de efecto conjunta.
- **Evaluación de la heterogeneidad**. Genera el *forest plot* y las medidas de heterogeneidad τ^2, I^2 y Q para comprobar como de homogéneos son los estudios entre sí.
- **Evaluación del sesgo de publicación**. Se produce el gráfico de embudo para estimar si existe algún tipo de sesgo de publicación.
- **Análisis de valores atípicos**. Se lleva a cabo un análisis de los valores más influyentes y posibles valores atípicos, generando un par de gráficas resumen.

Capítulo 4

Resultados

A lo largo de este capítulo se va a ofrecer una descripción de los resultados obtenidos en los distintos metaanálisis que se propusieron en el capítulo anterior. No se pretende realizar un análisis subjetivo, ni se va a entrar a valorar las acciones tomadas para obtener los resultados, eso se dejará para el Capítulo 5 de discusión de los resultados.

Así, se comenzará con un metaanálisis (uno por medida en realidad) aplicado a las 47 publicaciones que quedaron tras la fase de selección de estudios, se proseguirá con un metaanálisis estratificado por plataforma y finalmente se terminará con la inclusión de un segundo nivel de estratificación donde se considerará también el tipo de análisis usado para la detección de *malware*. Cabe destacar que solo se van a mostrar de forma exhaustiva todos los experimentos realizados en la primera sección. Aunque estos experimentos también se han llevado a cabo en las otras secciones, se ha decidido mostrar únicamente uno de ellos ya que el método escogido para el experimento no influye en el resultado final de forma excesiva. En el Apéndice A.2 se muestran las llamadas a la función usadas para llevar a cabo los distintos experimentos.

4.1. Análisis Global

En este primer bloque de experimentación, se va a realizar un metaanálisis de los 47 estudios de forma conjunta sin tomar en cuenta ninguna otra variable adicional. A pesar de ser 47 estudios, realmente se van a incluir 49 entradas en ambos metaanálisis, ya que uno de los estudios enfocado en Android prueba su modelo en dos datasets diferentes, y otra de las publicaciones prueba sus modelos por separado tanto en Windows como en Android. Se va a comenzar con el metaanálisis usando como medida la sensibilidad y se continuará con el aplicado a la especificidad.

4.1.1. Metaanálisis de proporciones aplicado a la sensibilidad

Se realizan cuatro experimentos distintos, cada uno de los cuales usará un método con una transformación específica: inverso de la varianza con el método logit, inverso de la varianza con el método FT, inverso de la varianza con el método arcoseno, y GLMM con el método logit. De este modo, se comprobará que no existe una diferencia relevante en los resultados obtenidos.

Experimento 1. Inverso de la varianza y logit

En primer lugar se presenta la estimación conjunta de la sensibilidad con su intervalo de confianza 95 %:

```
     pred    ci.lb    ci.ub    pi.lb    pi.ub
 0.985418 0.973725 0.991950 0.510522 0.999772
```

Como se observa, esta estimación toma el valor 0,985418 (o lo que es lo mismo una sensibilidad conjunta de aproximadamente un $98,54\,\%$) con un $I.C. = (0{,}973725, 0{,}991950)$.

El siguiente paso consistiría en el análisis de la heterogeneidad. Para ello se va a hacer uso tanto del diagrama de efectos de la Figura 4.1 como de los resultados ofrecidos por los test de heterogeneidad I^2, τ^2 y Q.

```
Random-Effects Model (k = 49; tau^2 estimator: DL)

tau^2 (estimated amount of total heterogeneity): 4.4352 (SE = 3.1101)
tau (square root of estimated tau^2 value):      2.1060
I^2 (total heterogeneity / total variability):   99.81%
H^2 (total variability / sampling variability):  536.18

Test for Heterogeneity:
Q(df = 48) = 25736.7822, p-val < .0001

Model Results:
estimate      se     zval    pval   ci.lb   ci.ub
  4.2133  0.3065  13.7458  <.0001  3.6125  4.8140  ***
---
Signif. codes:  0 "***" 0.001 "**" 0.01 "*"
                0.05 "." 0.1 " " 1

       estimate  ci.lb  ci.ub
tau^2      4.44   1.17   2.77
tau        2.11   1.08   1.66
I^2(%)    99.81  99.30  99.70
H^2      536.18 142.33 334.65
```

Los resultados de los test no ofrecen lugar a dudas de la existencia de una heterogeneidad muy alta. Basta observar los valores de I^2, que se encuentra cercano al 100 % con un intervalo de confianza al 95 % de (99,30, 99,70), y del estadístico Q, que se estima en 25736,7822 siendo claramente significativo ($p < 0{,}0001$).

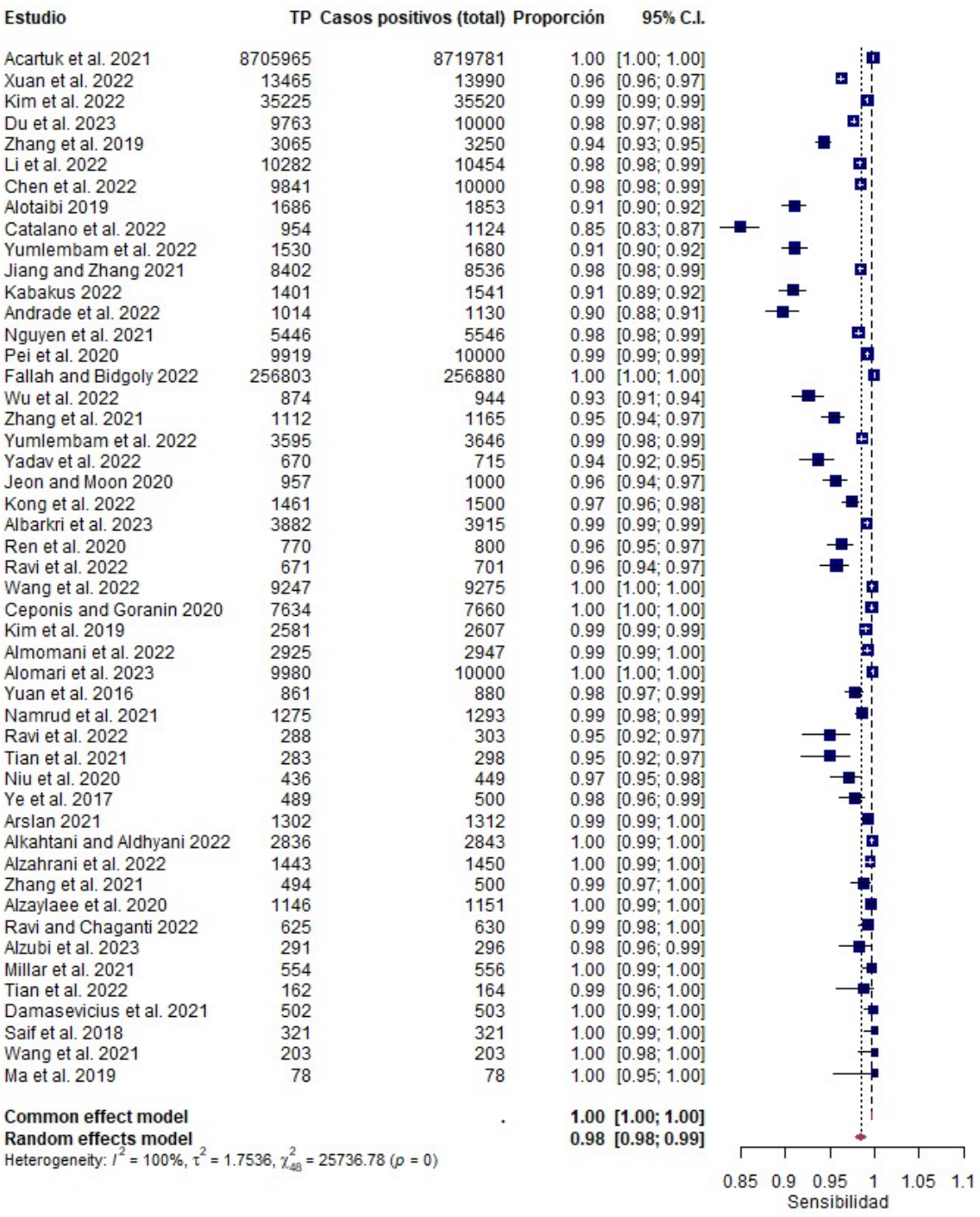

Figura 4.1: Experimento 1 - Diagrama de efectos para sensibilidad global

Al inspeccionar el diagrama de efectos también se advierte una heterogeneidad clara al existir múltiples estudios que no se cruzan con la estimación del modelo de efecto conjunto, que se sitúa en torno al 98 %. Se puede contemplar que el rango de valores que toma la sensibilidad de los estudios va desde el 85 % al 100 %, estando la mayor parte de ellos por encima del 95 %. Esta situación, junto con que los dos estudios de mayor tamaño (con gran diferencia sobre el resto) toman una sensibilidad cercana al 100 %, provoca que el valor medio estimado tome un valor tan alto. Además, el intervalo de confianza calculado para cada uno de los estudios, es en la mayor parte de los casos, muy estrecho. Ambas condiciones juegan a favor de la presencia de heterogeneidad. Posteriormente se analizarán los estudios influyentes y qué resultados se obtienen al eliminar aquellos que pueden ser atípicos.

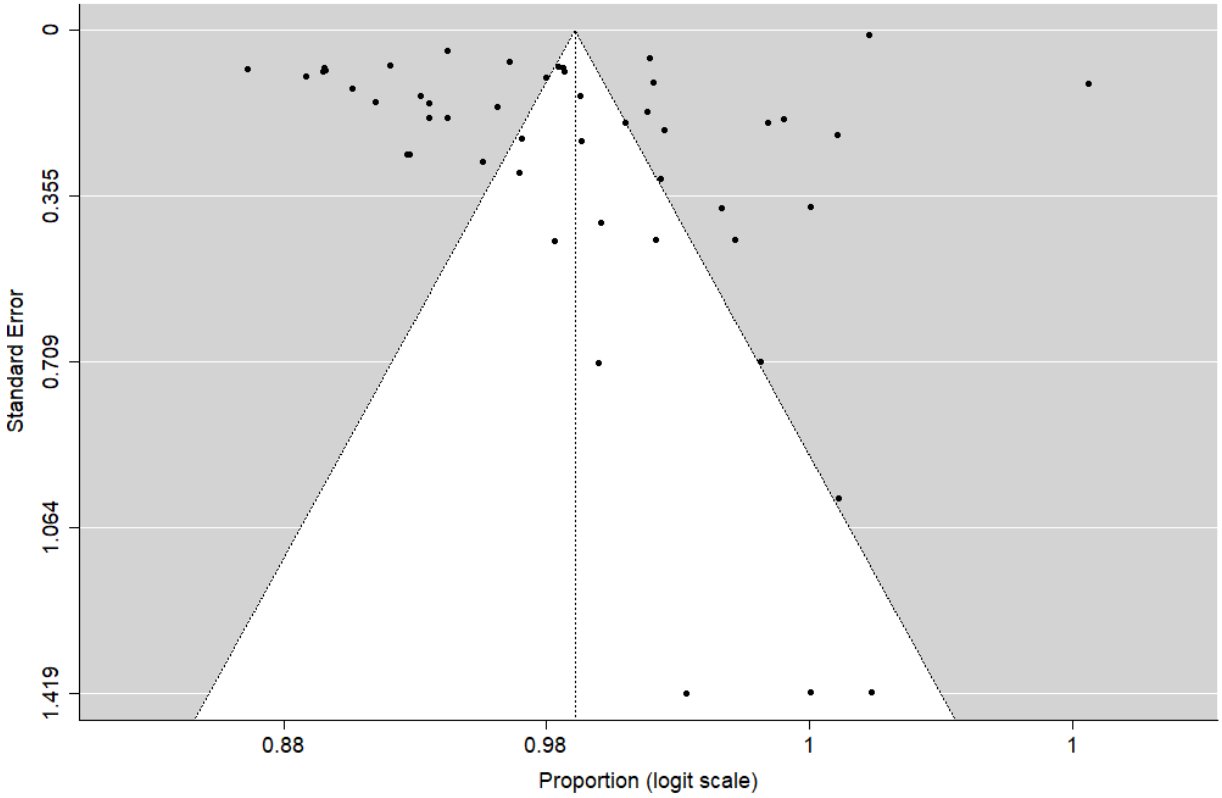

Figura 4.2: Experimento 1 - Diagrama de embudo para sensibilidad global

La siguiente cuestión a tratar será el sesgo de publicación. En la Sección 2.1.8, ya se comentó que el uso de la gráfica de embudo para mostrar el sesgo de publicación en los metaanálisis de proporciones no es aconsejable. No obstante, como ejemplo de lo que ocurre, se muestra el resultado ofrecido por esta gráfica en la Figura 4.2. A partir de los siguientes experimentos se va a omitir la exposición de esta gráfica ya que en todos los casos se obtienen resultados similares, confirmando lo indicado por Barker et al. [2021]. Analizando la figura se contempla la existencia de asimetría, sin embargo, dada la distribución de estudios en el

gráfico, no se puede asegurar que esta asimetría sea debida a sesgo de publicación por la ausencia de estudios pequeños.

Según Hunter et al. [2014], un enfoque alternativo para comprobar si la asimetría de la gráfica anterior es inducida por la forma en la que se construye la propia gráfica podría ser utilizar en el eje de las ordenadas el tamaño de la muestra en lugar del error estándar. En la Figura 4.3 se muestra este resultado. En este caso, la presencia de un estudio de tamaño muy superior al resto no permite observar con claridad si existen estudios pequeños, habría que excluirlo del análisis para comprobar el resultado, pero esto se llevará a cabo en siguientes pasos, ya que dicho estudio va a ser marcado como atípico.

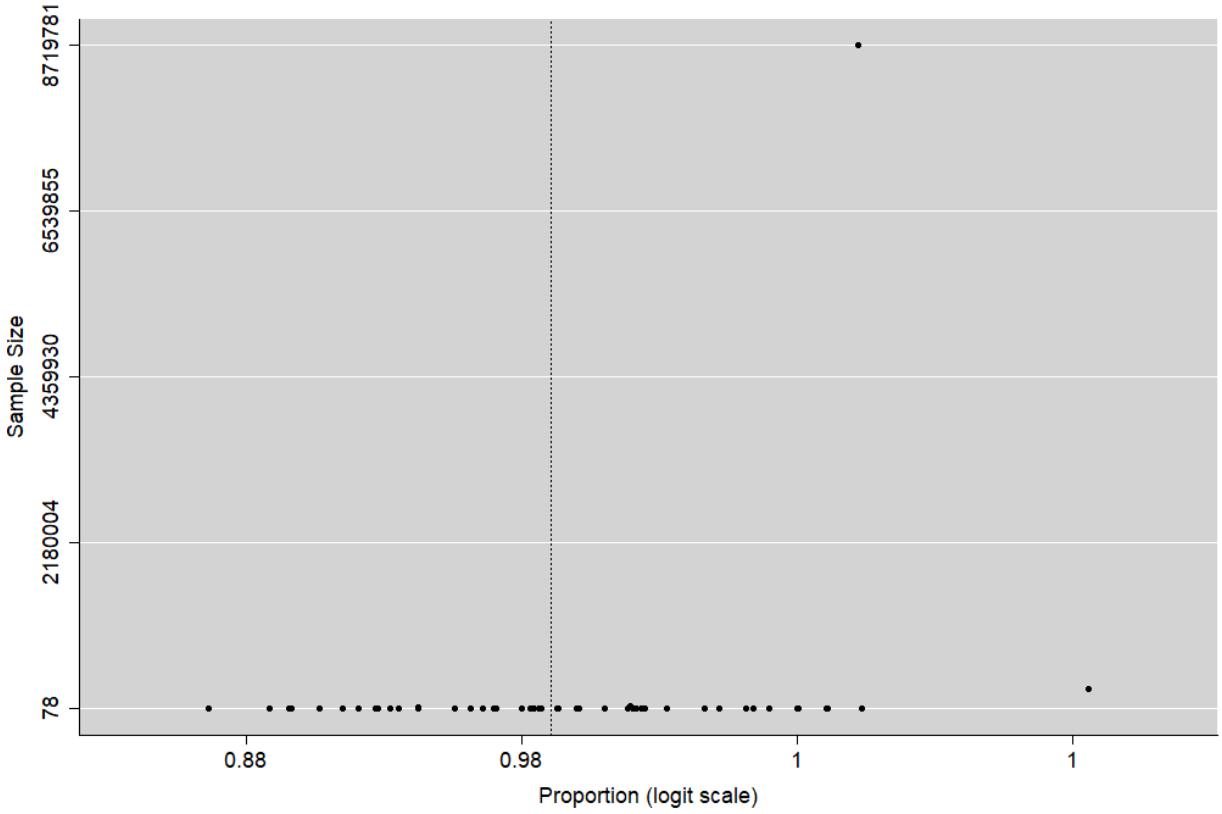

Figura 4.3: Experimento 1 - Diagrama de embudo con tamaño de muestra para sensibilidad global

El siguiente paso que se va a realizar, va a ser un análisis de los datos atípicos e influyentes para, en un último paso, evaluar si con su exclusión se mejoran los datos obtenidos. La Figura 4.4 muestra un diagrama de efectos donde se indica el peso que tiene la exclusión de cada uno de los estudios con respecto al valor estimado del efecto conjunto representado por la línea de referencia vertical. Así, los estudios más influyentes son aquellos que están más alejados de dicha línea. De este modo, se observa que el estudio Acarturk et al. [2021] es el más influyente y, además, viendo el tamaño del cuadrado que representa su estimación, es

también el estudio con la mayor muestra. Fallah and Bidgoly [2022] es otro posible estudio influyente.

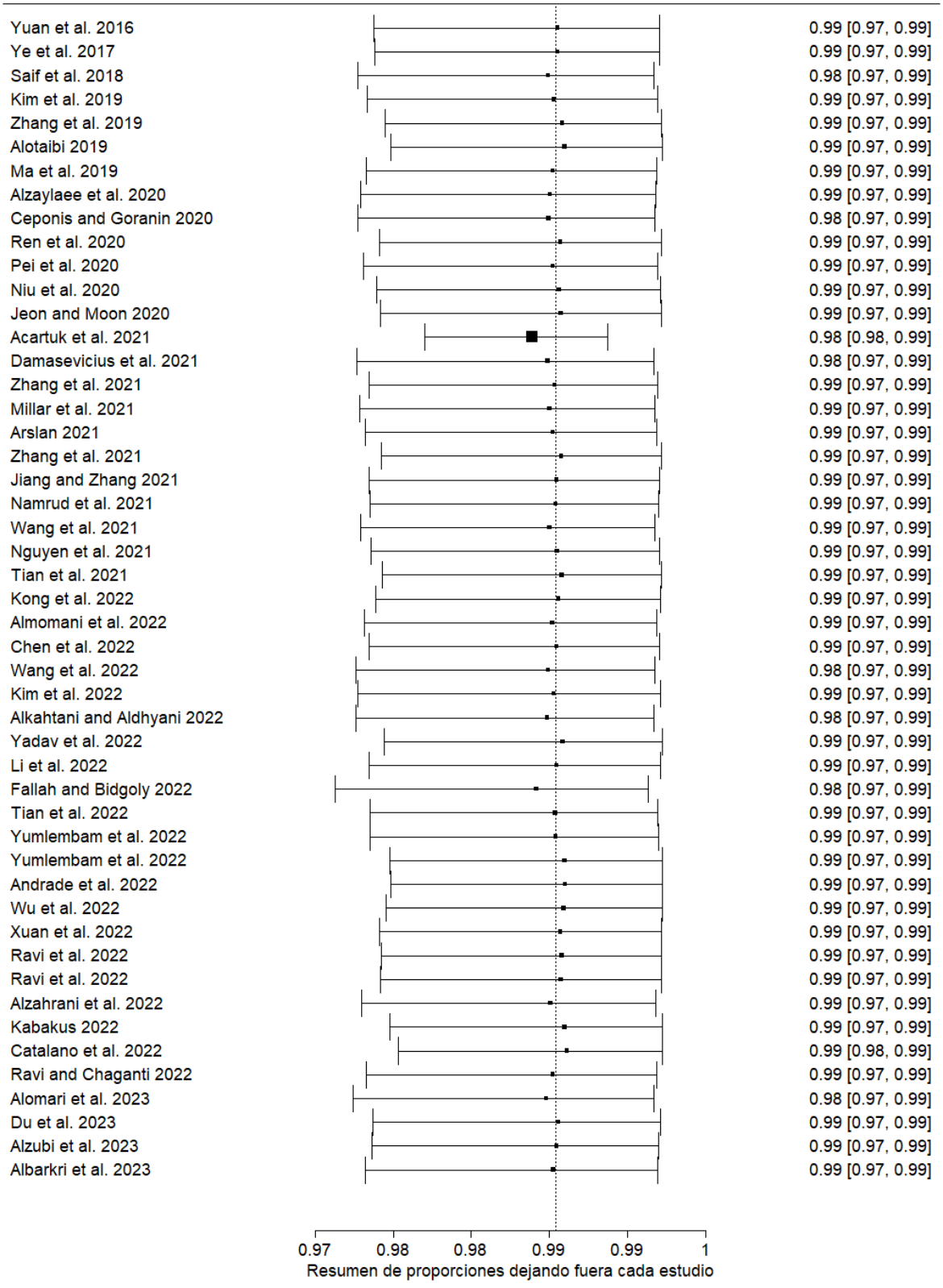

Figura 4.4: Experimento 1 - Diagrama de efectos con uno fuera para sensibilidad global

Para contrastar la información de esta gráfica se realiza un segundo análisis de influyentes que ofrece los resultados en la tabla de R y que se resumen en la Figura 4.5. Ahí se puede ver que el estudio 14, que se corresponde con Acarturk et al. [2021] es, en efecto, considerado como influyente.

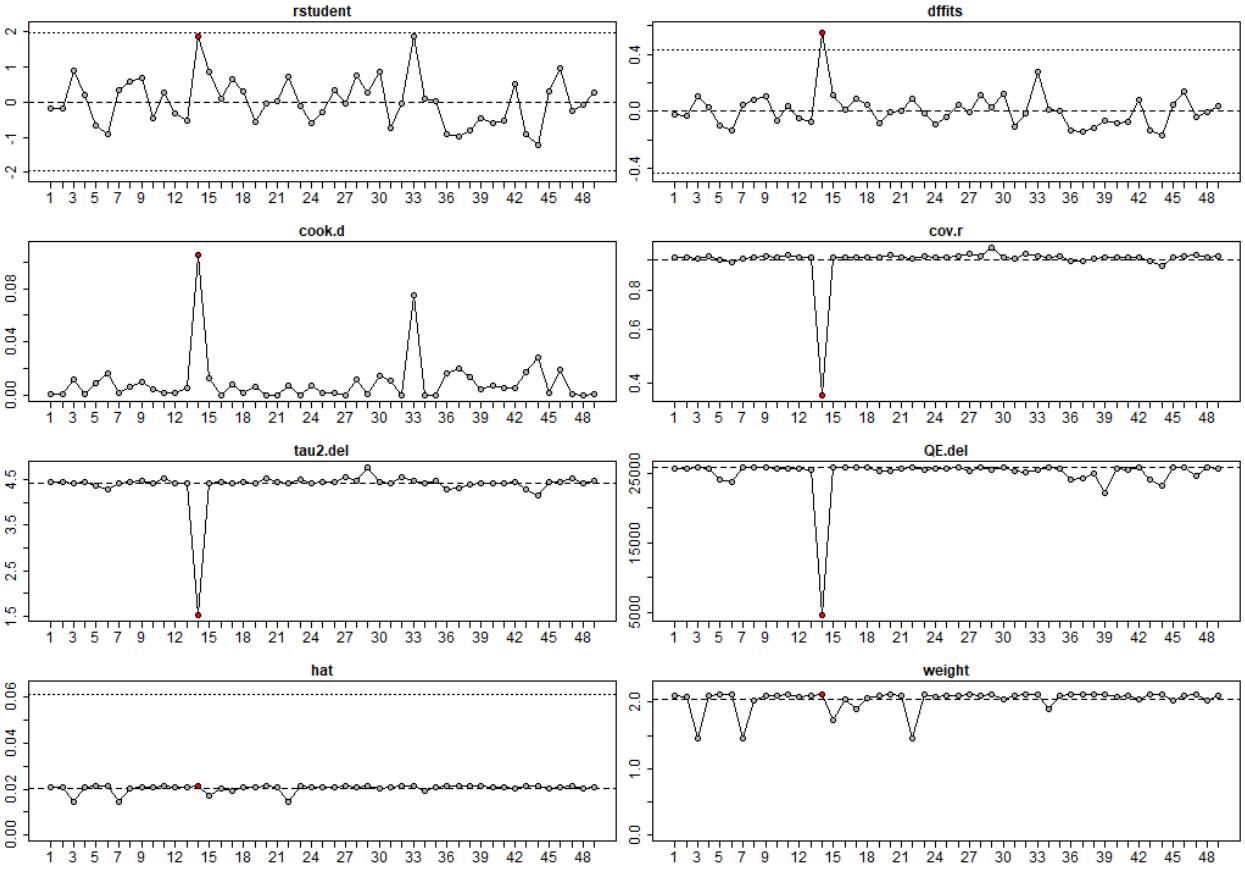

Figura 4.5: Experimento 1 - Estudios influyentes (marcados con puntos rojos) para sensibilidad global

	rstudent	dffits	cook.d	cov.r	tau2.del	QE.del	hat	weight	dfbs	inf
1	−0.1904	−0.0281	0.0008	1.0236	4.4450	25655.1658	0.0209	2.0929	−0.0281	
2	−0.1988	−0.0291	0.0008	1.0224	4.4408	25688.7030	0.0207	2.0748	−0.0291	
3	0.8944	0.1088	0.0118	1.0150	4.4360	25736.6266	0.0146	1.4592	0.1088	
4	0.1831	0.0262	0.0007	1.0271	4.4607	25692.5416	0.0210	2.0999	0.0262	
5	−0.6797	−0.0980	0.0095	1.0049	4.3604	24042.3338	0.0212	2.1155	−0.0980	
6	−0.9276	−0.1324	0.0169	0.9868	4.2791	23754.4979	0.0212	2.1151	−0.1323	
7	0.3344	0.0406	0.0017	1.0150	4.4360	25736.4219	0.0146	1.4570	0.0406	
8	0.5726	0.0822	0.0068	1.0221	4.4416	25735.6670	0.0203	2.0265	0.0822	
9	0.6992	0.1016	0.0104	1.0288	4.4684	25735.4618	0.0210	2.1000	0.1016	
10	−0.4627	−0.0678	0.0046	1.0218	4.4367	25531.7198	0.0210	2.1019	−0.0678	
11	0.2818	0.0392	0.0016	1.0411	4.5226	25639.0907	0.0211	2.1123	0.0392	
12	−0.3331	−0.0487	0.0024	1.0221	4.4390	25664.3104	0.0208	2.0811	−0.0487	
13	−0.5318	−0.0779	0.0061	1.0207	4.4316	25412.1659	0.0211	2.1067	−0.0779	
14	1.8737	0.5548	0.1051	0.3657	1.5147	4627.0599	0.0212	2.1182	0.5422	*
15	0.8674	0.1150	0.0132	1.0179	4.4367	25736.6858	0.0173	1.7279	0.1150	
16	0.0929	0.0133	0.0002	1.0220	4.4407	25723.4918	0.0204	2.0407	0.0133	
17	0.6409	0.0892	0.0080	1.0200	4.4379	25736.6217	0.0190	1.9030	0.0892	
18	0.3107	0.0449	0.0020	1.0236	4.4462	25726.0679	0.0207	2.0712	0.0449	
19	−0.5605	−0.0821	0.0067	1.0199	4.4277	25320.5277	0.0211	2.1089	−0.0821	
20	−0.0359	−0.0077	0.0001	1.0438	4.5347	25320.5930	0.0211	2.1147	−0.0077	
21	0.0224	0.0029	0.0000	1.0246	4.4497	25688.5550	0.0209	2.0917	0.0029	
22	0.7127	0.0867	0.0075	1.0150	4.4361	25736.7771	0.0146	1.4588	0.0867	
23	−0.1029	−0.0167	0.0003	1.0360	4.4996	25376.3304	0.0211	2.1134	−0.0167	
24	−0.6077	−0.0886	0.0078	1.0205	4.4318	25610.9862	0.0209	2.0853	−0.0886	

```
25  −0.2819 −0.0417 0.0017 1.0247   4.4495 25538.0880 0.0211 2.1058 −0.0417
26   0.3221  0.0465 0.0022 1.0268   4.4594 25714.1318 0.0210 2.0966  0.0465
27  −0.0420 −0.0090 0.0001 1.0478   4.5525 25235.0481 0.0212 2.1152 −0.0090
28   0.7552  0.1097 0.0121 1.0295   4.4711 25736.4562 0.0210 2.1013  0.1097
29   0.2623  0.0308 0.0010 1.0960   4.7686 25359.8434 0.0212 2.1166  0.0308
30   0.8449  0.1221 0.0149 1.0230   4.4443 25736.7173 0.0205 2.0520  0.1221
31  −0.7257 −0.1059 0.0112 1.0167   4.4136 25301.9546 0.0211 2.1070 −0.1059
32  −0.0585 −0.0116 0.0001 1.0491   4.5585 25172.5467 0.0212 2.1154 −0.0116
33   1.8613  0.2725 0.0748 1.0291   4.4690 25361.0147 0.0211 2.1121  0.2725
34   0.0823  0.0114 0.0001 1.0198   4.4371 25732.2569 0.0190 1.9013  0.0114
35   0.0200  0.0019 0.0000 1.0307   4.4761 25599.0773 0.0211 2.1088  0.0019
36  −0.9206 −0.1318 0.0168 0.9909   4.2977 23966.4206 0.0211 2.1148 −0.1318
37  −0.9940 −0.1429 0.0199 0.9938   4.3109 24272.0018 0.0211 2.1137 −0.1429
38  −0.8132 −0.1182 0.0138 1.0111   4.3883 24992.1122 0.0211 2.1109 −0.1182
39  −0.4648 −0.0684 0.0047 1.0218   4.4358 22042.6650 0.0212 2.1173 −0.0684
40  −0.5993 −0.0874 0.0076 1.0206   4.4320 25612.3563 0.0209 2.0853 −0.0874
41  −0.5287 −0.0774 0.0060 1.0210   4.4329 25511.2244 0.0210 2.1018 −0.0774
42   0.5261  0.0759 0.0058 1.0229   4.4439 25734.4437 0.0205 2.0519  0.0759
43  −0.9291 −0.1332 0.0172 0.9924   4.3045 24070.5935 0.0211 2.1145 −0.1332
44  −1.2299 −0.1738 0.0284 0.9619   4.1678 23190.5387 0.0211 2.1150 −0.1737
45   0.2884  0.0413 0.0017 1.0219   4.4407 25730.9999 0.0203 2.0262  0.0413
46   0.9513  0.1385 0.0193 1.0271   4.4606 25734.9261 0.0209 2.0946  0.1385
47  −0.2350 −0.0371 0.0014 1.0453   4.5415 24612.2147 0.0212 2.1162 −0.0371
48  −0.0701 −0.0102 0.0001 1.0215   4.4388 25720.0638 0.0203 2.0254 −0.0102
49   0.2640  0.0378 0.0014 1.0292   4.4700 25694.1571 0.0210 2.1038  0.0378
```

El último paso consiste entonces en realizar el mismo análisis eliminando este estudio influyente de modo que si se mantiene la heterogeneidad alta y aparecen nuevos estudios influyentes se procedería a eliminarlos también y así sucesivamente hasta que, o bien se obtenga una heterogeneidad mucho menor de la actual o bien no haya estudios que puedan ser excluidos según este criterio. Aquí, para no sobrecargar el trabajo de forma innecesaria, se va a mostrar únicamente el resultado final de todo este proceso y a comentar brevemente. Tras la eliminación del estudio Acarturk et al. [2021], se observa que el estudio 33, Fallah and Bidgoly [2022], también es influyente, por lo que se excluye también. Tras esta segunda eliminación, no aparecen más estudios influyentes y los resultados obtenidos no cambian mucho, como se observará a continuación.

La sensibilidad conjunta estimada no varía excesivamente, pasa del 0,985418 al 0,981919, aunque su intervalo de confianza al 95 % se ha estrechado ligeramente como se observa en este primer resultado:

```
1      pred     ci.lb     ci.ub     pi.lb     pi.ub
2  0.981919  0.975646  0.986599  0.880842  0.997500
```

En cuanto a la heterogeneidad tampoco hay grandes cambios, se reducen un poco los valores del estadístico Q y de I^2, aunque esta última sigue estando próxima al 100 %, por lo que la heterogeneidad sigue siendo muy alta.

```
1 Random-Effects Model (k = 47; tau^2 estimator: DL)
2
3 tau^2 (estimated amount of total heterogeneity): 1.0112 (SE = 0.3577)
4 tau (square root of estimated tau^2 value):      1.0056
5 I^2 (total heterogeneity / total variability):   98.48%
6 H^2 (total variability / sampling variability):  65.71
7
8 Test for Heterogeneity:
```

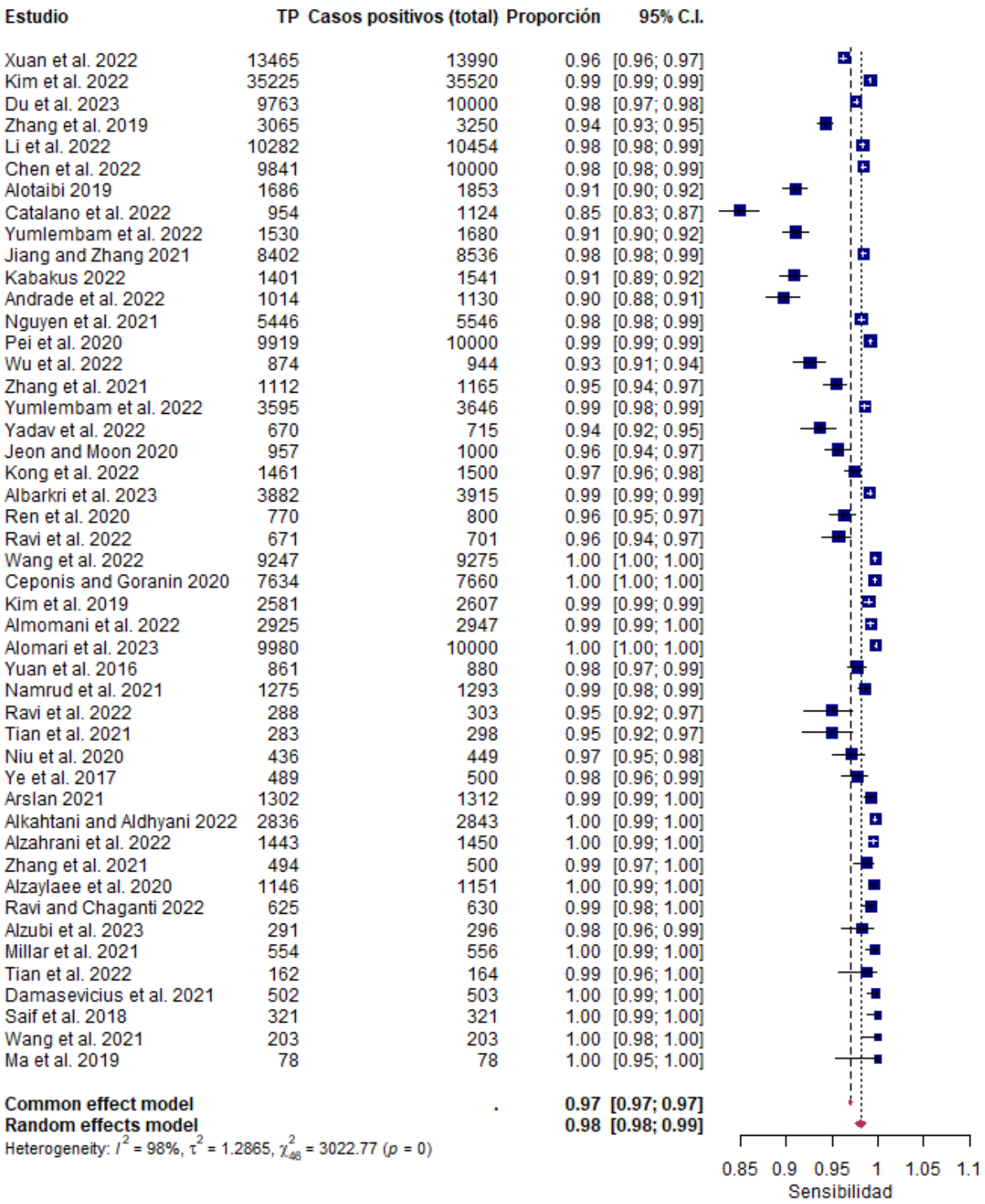

Figura 4.6: Experimento 1b - Diagrama de efectos con 2 estudios excluidos para sensibilidad global

```
Q(df = 46) = 3022.7657, p-val < .0001

Model Results:
```

```
estimate      se     zval    pval   ci.lb   ci.ub
  3.9947  0.1552  25.7316  <.0001  3.6904  4.2989  ***
---
Signif. codes:  0 "***" 0.001 "**" 0.01 "*" 0.05 "."
                0.1 " " 1

       estimate ci.lb  ci.ub
tau^2      1.01  0.86   2.15
tau        1.01  0.93   1.47
I^2(%)    98.48 98.21  99.28
H^2       65.71 55.90 138.40
```

La Figura 4.6 muestra que el diagrama de efectos es muy similar al obtenido incluyendo los dos estudios influyentes.

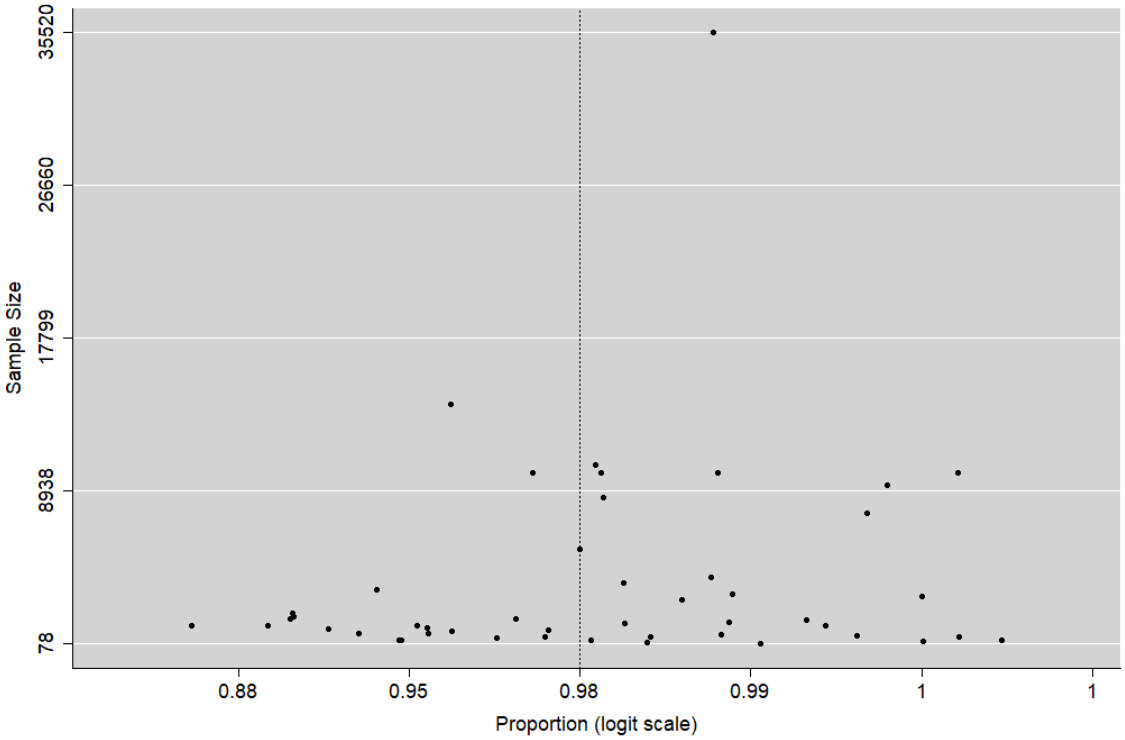

Figura 4.7: Experimento 1b - Diagrama de embudo con tamaño de muestra y 2 estudios excluidos para sensibilidad global

Respecto al sesgo, ahora sí se observa claramente la presencia de estudios pequeños, así como la existencia de cierta asimetría como se puede ver en la Figura 4.7

Por último, en la Figura 4.8 se observa que ya no existen más estudios influyentes, al no haber ningún punto rojo. A pesar de esto, se han realizado diversas pruebas en las que se excluyen estudios que en el diagrama de efectos puedan están alejados del estimador conjunto

como puede ser el estudio Catalano et al. [2022] y, sin tener en cuenta la validez o no de esta exclusión, no se ha conseguido reducir la heterogeneidad en términos razonables, por lo que estos resultados no van a ser presentados en este trabajo.

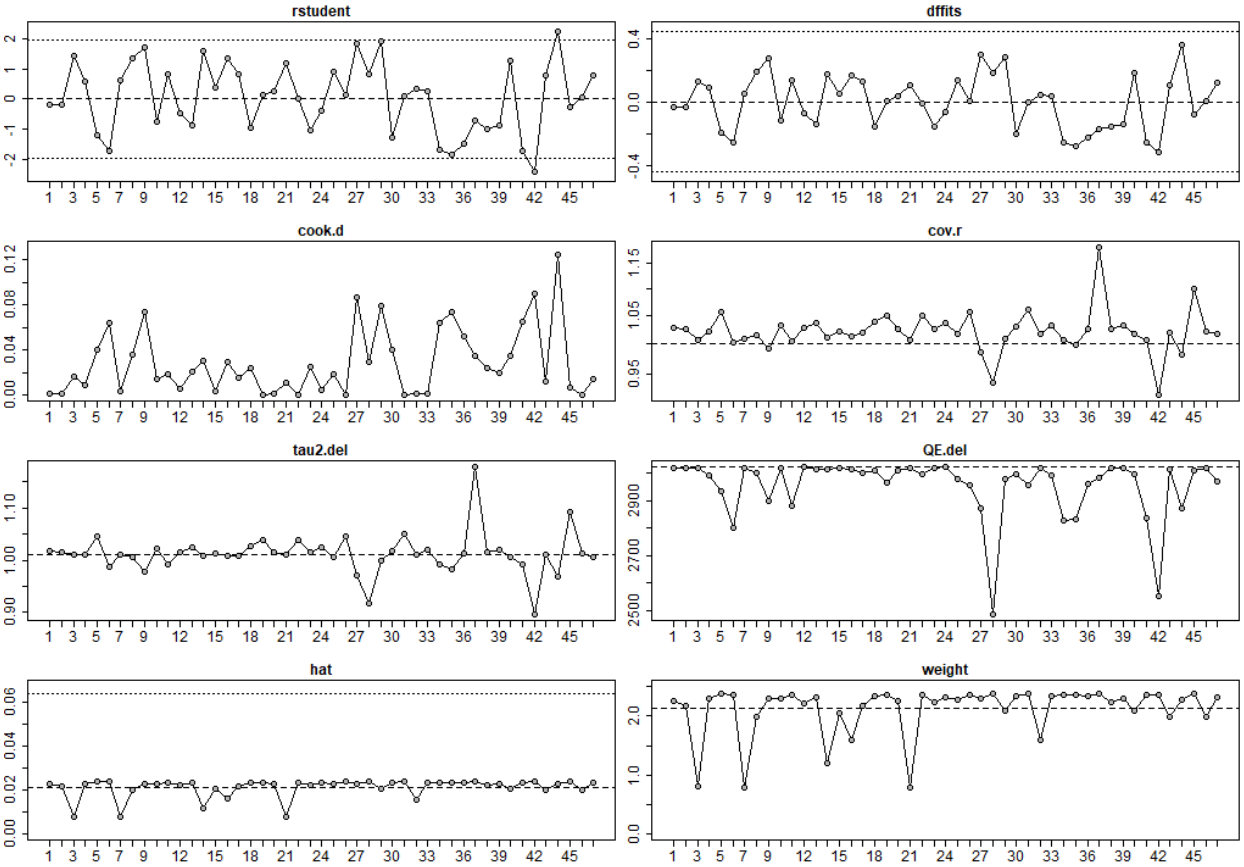

Figura 4.8: Experimento 1b - Estudios influyentes (marcados con puntos rojos) con 2 estudios excluidos para sensibilidad global

Experimento 2. Inverso de la varianza y FT

Tanto en este experimento como en los dos siguientes se evalúa si la heterogeneidad que ha aparecido en el experimento anterior está relacionada con el modelo o la transformación utilizada. En esta ocasión no habrá cambio en el modelo pero se utilizará la transformación Freeman-Tukey.

La sensibilidad conjunta estimada es algo menor que la calculada en el anterior experimento, siendo en este caso ligeramente inferior al 98 %:

```
1  pred     ci.lb     ci.ub     pi.lb     pi.ub
2  0.979532 0.975129 0.983526 0.942398 0.998327
```

La heterogeneidad sigue siendo muy alta, siendo el valor de I^2 superior al 99 %:

```
1 Random-Effects Model (k = 49; tau^2 estimator: DL)
```

```
2
3 tau^2 (estimated amount of total heterogeneity): 0.0024 (SE = 0.0025)
4 tau (square root of estimated tau^2 value):      0.0494
5 I^2 (total heterogeneity / total variability):   99.41%
6 H^2 (total variability / sampling variability):  170.77
7
8 Test for Heterogeneity:
9 Q(df = 48) = 8196.8022, p-val < .0001
10
11 Model Results:
12 estimate      se      zval    pval   ci.lb   ci.ub
13   1.4250  0.0074  191.4550  <.0001  1.4104  1.4395  ***
14 ---
15 Signif. codes:  0 "***" 0.001 "**" 0.01 "*" 0.05 "."
16                 0.1 " " 1
17
18        estimate  ci.lb  ci.ub
19 tau^2      0.00   0.01   0.01
20 tau        0.05   0.07   0.11
21 I^2(%)    99.41  99.73  99.88
22 H^2      170.77 376.62 868.46
```

Por otro lado, los estudios marcados como influyentes no son los mismos que en el experimento anterior. En este caso se parte de cinco estudios influyentes en la primera iteración y se llega hasta los once conforme se van excluyendo. Sin embargo, la exclusión, ya sea parcial o total, de este conjunto de estudios tampoco mejora ninguno de los parámetros del metaanálisis forma sustancial.

Experimento 3. Inverso de la varianza y arcoseno

En esta ocasión, la modificación realizada respecto al experimento original consiste en usar la transformación arcoseno en lugar de la logit.

Los resultados obtenidos son prácticamente los mismos que los obtenidos en el experimento anterior:

```
1     pred    ci.lb    ci.ub    pi.lb    pi.ub
2  0.979780 0.975482 0.983672 0.943398 0.997963
```

```
1 Random-Effects Model (k = 49; tau^2 estimator: DL)
2
3 tau^2 (estimated amount of total heterogeneity): 0.0024 (SE = 0.0025)
4 tau (square root of estimated tau^2 value):      0.0492
5 I^2 (total heterogeneity / total variability):   99.41%
```

```
H^2 (total variability / sampling variability):  169.75

Test for Heterogeneity:
Q(df = 48) = 8148.1775, p-val < .0001

Model Results:
estimate      se      zval    pval   ci.lb   ci.ub
  1.4281  0.0074  192.3862  <.0001  1.4136  1.4427  ***
---
Signif. codes:  0 "***" 0.001 "**" 0.01 "*" 0.05 "." 0.1 " " 1

       estimate  ci.lb  ci.ub
tau^2      0.00   0.01   0.01
tau        0.05   0.08   0.12
I^2(%)    99.41  99.75  99.89
H^2      169.75 403.88 933.34
```

En cuanto a los estudios que se marcan como influyentes, coinciden con los del experimento 2 salvo que en esta ocasión el estudio Fallah and Bidgoly [2022] no es influyente. De nuevo la exclusión parcial o total de estos estudios no produce mejoras en cuanto a la heterogeneidad.

Experimento 4. GLMM y logit

La evaluación llevada a cabo en este experimento requiere de un cambio de modelo, se pasa del Inverso de la Varianza a GLMM. En este caso sí se mantiene la transformación logit.

Este modelo ofrece un valor de la estimación conjunta de la sensibilidad muy similar al obtenido en el experimento 1:

```
     pred    ci.lb    ci.ub    pi.lb    pi.ub
 0.986052 0.979390 0.990581 0.824611 0.999060
```

Por otra parte, adolece de los mismos problemas que el resto de experimentos, aunque en esta ocasión no se calcula el estimador Q, sí se puede ver que el valor de I^2 sigue por encima del 99 %, lo que indica una clara heterogeneidad.

```
Random-Effects Model (k = 49; tau^2 estimator: ML)

tau^2 (estimated amount of total heterogeneity): 1.8714
tau (square root of estimated tau^2 value):      1.3680
I^2 (total heterogeneity / total variability):   99.56%
H^2 (total variability / sampling variability):  226.81

```

```
Tests for Heterogeneity:
Wld(df = 48) = 25736.2612, p-val < .0001
LRT(df = 48) = 13432.6766, p-val < .0001

Model Results:
estimate      se     zval    pval   ci.lb   ci.ub
  4.2584  0.2027  21.0112  <.0001  3.8611  4.6556  ***
---
Signif. codes:  0 "***" 0.001 "**" 0.01 "*" 0.05 "."
                0.1 " " 1
```

Aunque se ha probado a excluir los estudios que aparecían en anteriores experimentos no se ha encontrado ninguna mejora significativa.

4.1.2. Metaanálisis de proporciones aplicado a la especificidad

De aquí en adelante no se van a presentar todos los experimentos realizados modificando modelos y transformaciones, ya que los resultados obtenidos en todos los casos son muy parecidos entre sí. Salvo que se diga lo contrario, en el resto del trabajo se va a utilizar el modelo de inverso de la varianza con la transformación Freeman-Tukey.

La estimación conjunta de la especificidad con su intervalo de confianza al 95 % es 0,976382 (o lo que es lo mismo una especificidad conjunta de aproximadamente un 97,64 %) tal y como se observa en el resultado de R:

```
    pred    ci.lb    ci.ub    pi.lb    pi.ub
0.976382 0.969160 0.982690 0.909276 1.000000
```

En cuanto a los test y medidas de heterogeneidad, analizando tanto los resultados numéricos ofrecidos por R como el diagrama de efectos indicado de la Figura 4.9 se observa una heterogeneidad muy alta. Al igual que ocurría en el caso de la sensibilidad, también existen muchos estudios que tienen un intervalo de confianza del estimador de su especificidad muy pequeño. Esto provoca que la capacidad de un estudio de cruzarse con la línea vertical que representa la estimación conjunta sea muy baja, lo que aumenta la heterogeneidad.

Los resultados de los test confirman una heterogeneidad muy alta ya que el valor de I^2, se encuentra cercano al 100 %, en concreto en el 99,77 %, con un intervalo de confianza al 95 % de (99,68, 99,86), y el estadístico Q, que se estima en 20519,3192 siendo claramente significativo $p < 0{,}0001$.

```
Random-Effects Model (k = 49; tau^2 estimator: DL)

tau^2 (estimated amount of total heterogeneity): 0.0058 (SE = 0.0052)
tau (square root of estimated tau^2 value):      0.0759
I^2 (total heterogeneity / total variability):   99.77%
```

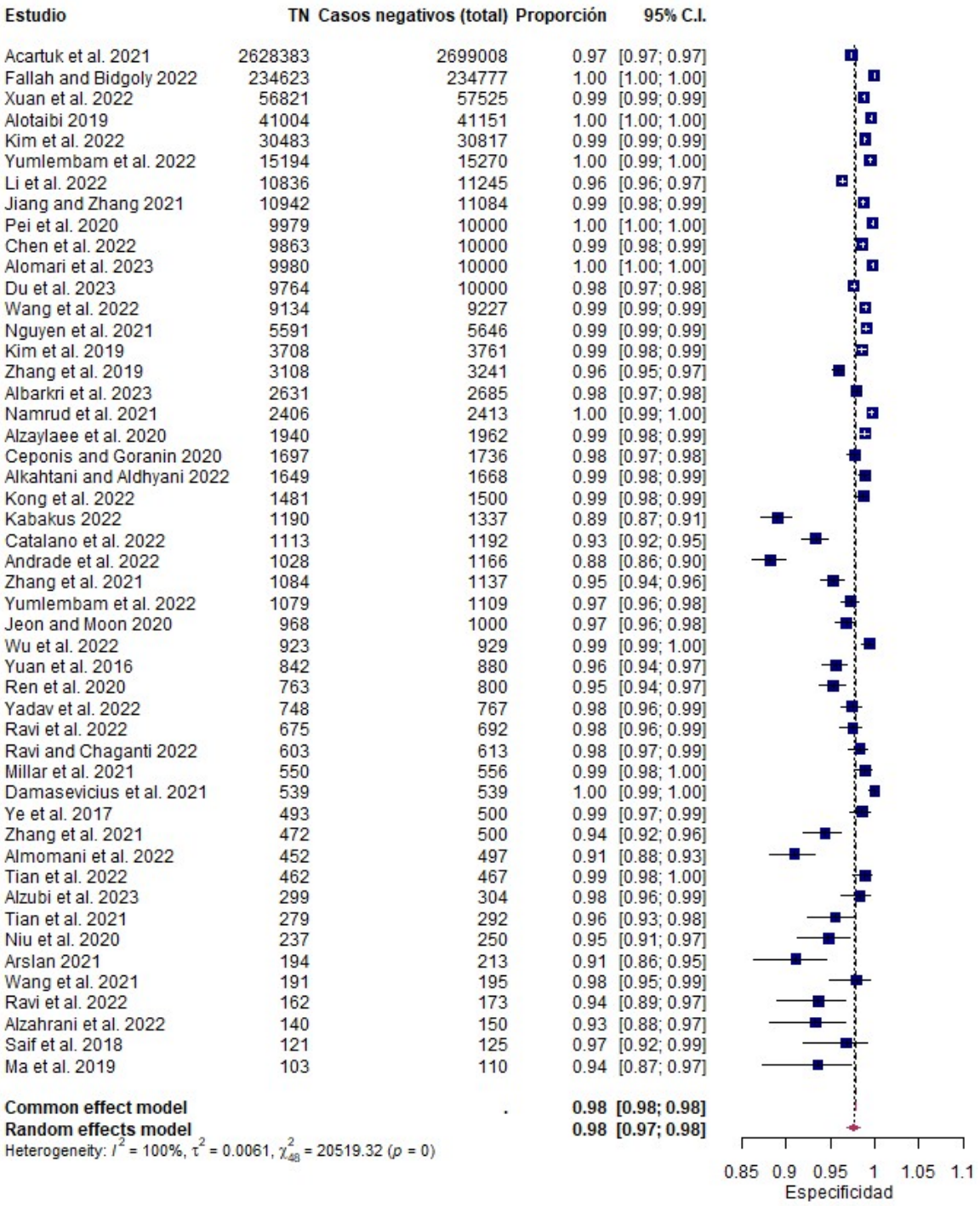

Figura 4.9: Experimento 5 - Diagrama de efectos para especificidad global

```
H^2 (total variability / sampling variability):  427.49

Test for Heterogeneity:
Q(df = 48) = 20519.3192, p-val < .0001

Model Results:
```

```
estimate       se       zval     pval    ci.lb    ci.ub
  1.4140   0.0112   126.4976   <.0001   1.3921   1.4359   ***
---
Signif. codes:  0 "***" 0.001 "**" 0.01 "*" 0.05 "."
                0.1 " " 1

        estimate  ci.lb  ci.ub
tau^2       0.01   0.00   0.01
tau         0.08   0.06   0.10
I^2(%)     99.77  99.68  99.86
H^2       427.49 309.86 728.30
```

Como se ha mencionado anteriormente, el sesgo no se va a poder analizar con el gráfico del embudo ni tampoco con la alternativa propuesta, por lo que lo único que quedaría por revisar sería la existencia de elementos influyentes o valores atípicos. Según la información ofrecida por R, no existen influyentes al utilizar todos los estudios (no hay ningún '*' en la última columna 'inf'):

```
    rstudent  dffits cook.d  cov.r tau2.del     QE.del    hat weight    dfbs inf
1    -0.6991 -0.1014 0.0103 1.0222   0.0058 20505.4858 0.0207 2.0686 -0.1014
2     0.4367  0.0625 0.0039 1.0213   0.0058 20518.0334 0.0200 1.9973  0.0625
3    -0.3784 -0.0484 0.0023 1.0166   0.0058 20518.4376 0.0161 1.6126 -0.0484
4     0.4922  0.0743 0.0056 1.0285   0.0058 20507.1721 0.0215 2.1458  0.0743
5    -0.6278 -0.0920 0.0085 1.0261   0.0058 20478.0173 0.0214 2.1419 -0.0920
6     1.2796  0.1940 0.0383 1.0394   0.0059 19227.1380 0.0217 2.1683  0.1941
7    -1.1933 -0.1501 0.0225 1.0158   0.0058 20513.4851 0.0156 1.5582 -0.1501
8     0.6518  0.0967 0.0094 1.0248   0.0058 20506.3253 0.0212 2.1236  0.0967
9     0.0713  0.0112 0.0001 1.0249   0.0058 20519.2369 0.0212 2.1176  0.0112
10   -0.7947 -0.1150 0.0132 1.0218   0.0058 20503.5541 0.0206 2.0589 -0.1150
11    1.4687  0.2181 0.0475 1.0213   0.0058 20105.4796 0.0216 2.1612  0.2181
12   -0.9482 -0.1301 0.0169 1.0190   0.0058 20511.8948 0.0185 1.8499 -0.1301
13   -0.3175 -0.0459 0.0021 1.0230   0.0058 20514.8962 0.0208 2.0803 -0.0459
14   -0.1099 -0.1345 0.0122 0.7053   0.0039  5713.2232 0.0217 2.1706 -0.1328
15    1.7332  0.2480 0.0615 1.0199   0.0058 20484.8300 0.0201 2.0089  0.2480
16   -1.0730 -0.1531 0.0235 1.0205   0.0058 20502.0428 0.0200 1.9973 -0.1531
17    0.6235  0.0896 0.0080 1.0215   0.0058 20515.7970 0.0201 2.0135  0.0896
18   -1.8170 -0.2464 0.0607 1.0178   0.0058 20497.8117 0.0180 1.8037 -0.2464
19   -0.8069 -0.1177 0.0139 1.0225   0.0058 20496.6167 0.0209 2.0908 -0.1177
20    0.5693  0.0885 0.0080 1.0408   0.0059 20466.9513 0.0216 2.1621  0.0886
21    1.3340  0.1971 0.0389 1.0225   0.0058 20437.1986 0.0213 2.1322  0.1971
22    0.0592  0.0080 0.0001 1.0185   0.0058 20519.3070 0.0178 1.7760  0.0080
23    0.7603  0.1145 0.0132 1.0302   0.0058 20465.8049 0.0215 2.1540  0.1145
24   -0.7387 -0.1025 0.0105 1.0196   0.0058 20513.8502 0.0189 1.8900 -0.1025
25    0.5585  0.0825 0.0068 1.0241   0.0058 20512.4939 0.0211 2.1095  0.0825
26   -1.9200 -0.2743 0.0752 1.0190   0.0057 20468.9177 0.0200 1.9963 -0.2743
27    0.5181  0.0806 0.0066 1.0394   0.0059 20482.2158 0.0216 2.1612  0.0807
28    0.7393  0.1127 0.0129 1.0357   0.0058 20437.1891 0.0216 2.1604  0.1127
29    0.6824  0.1114 0.0130 1.0715   0.0061 20283.1330 0.0217 2.1675  0.1115
30    0.6371  0.0942 0.0089 1.0243   0.0058 20508.8273 0.0212 2.1155  0.0942
31   -0.0416 -0.0057 0.0000 1.0224   0.0058 20518.8710 0.0205 2.0543 -0.0057
32   -0.4635 -0.0651 0.0043 1.0394   0.0059 20431.6608 0.0216 2.1622 -0.0652
33    2.5024  0.0931 0.0041 0.5094   0.0027  5363.0586 0.0217 2.1702  0.0906
34    0.6147  0.0877 0.0077 1.0210   0.0058 20516.4124 0.0199 1.9861  0.0877
35   -0.1269 -0.0181 0.0003 1.0234   0.0058 20517.7843 0.0209 2.0888 -0.0181
36    1.1373  0.1716 0.0298 1.0338   0.0058 20154.5164 0.0216 2.1644  0.1716
37   -2.5583 -0.3756 0.1401 1.0144   0.0057 20325.4994 0.0209 2.0927 -0.3756
38    0.9519  0.1388 0.0193 1.0222   0.0058 20503.9452 0.0207 2.0737  0.1388
39    0.5840  0.1062 0.0125 1.1257   0.0064 20197.8187 0.0217 2.1690  0.1065
40   -1.2265 -0.1630 0.0266 1.0176   0.0058 20510.6113 0.0174 1.7360 -0.1630
41   -0.0358 -0.0049 0.0000 1.0222   0.0058 20518.9439 0.0204 2.0425 -0.0049
42   -1.2880 -0.1686 0.0284 1.0171   0.0058 20510.8155 0.0168 1.6845 -0.1686
43   -2.3896 -0.3517 0.1229 1.0148   0.0057 20325.0557 0.0210 2.1023 -0.3517
44   -1.3652 -0.1998 0.0399 1.0208   0.0058 20458.2073 0.0209 2.0943 -0.1998
45    0.3304  0.0478 0.0023 1.0219   0.0058 20518.6246 0.0203 2.0271  0.0478
46    1.4836  0.2202 0.0484 1.0209   0.0057 20096.5277 0.0216 2.1612  0.2202
47    0.0323  0.0088 0.0001 1.0411   0.0059 20517.6457 0.0216 2.1612  0.0089
```

```
48    0.2768   0.0387  0.0015  1.0200    0.0058  20519.1012  0.0190  1.8997   0.0387
49    0.1830   0.0281  0.0008  1.0269    0.0058  20519.0501  0.0214  2.1360   0.0281
```

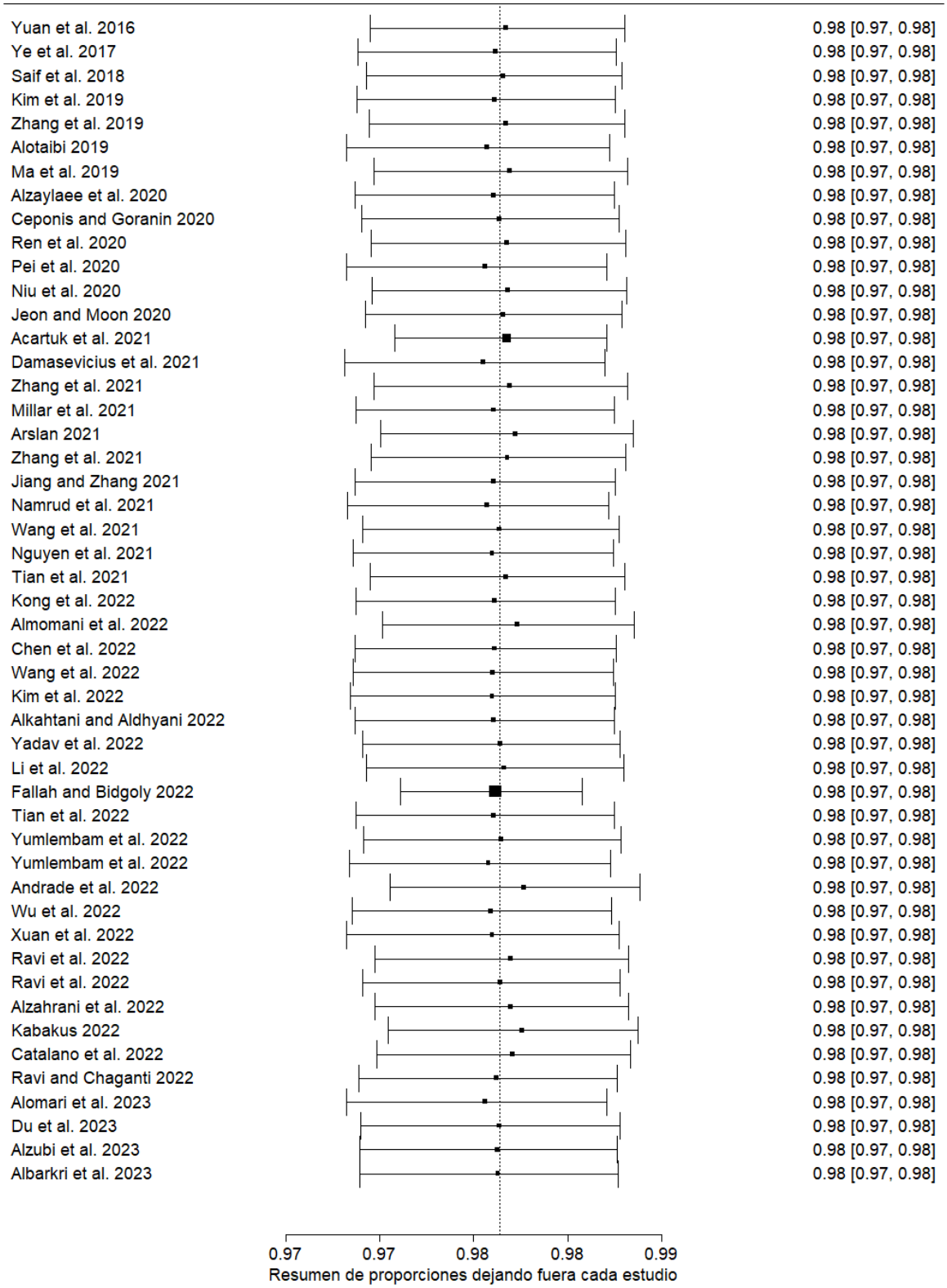

Figura 4.10: Experimento 5 - Diagrama de efectos con uno fuera para especificidad global

A pesar de ello, se ha procedido a eliminar alguno de los estudios que podrían dar más problemas teniendo en cuenta los resultados obtenidos en la Figura 4.10 al aplicar el método *leave-one-out*. Tras la eliminación de dichos estudios se han realizado las iteraciones necesarias para que desaparecieran los estudios influyentes y se ha ido revisando, tras cada paso, si la heterogeneidad bajaba. Los resultados en este sentido han sido infructuosos, ya que el valor de I^2 se ha mantenido más o menos estable incluso tras la supresión de una decena de estudios. De este modo, se ha decidido dejar el experimento de supresión de artículos ahí, para pasar a dividir los datos en subgrupos y realizar un metaanálisis de proporciones estratificado, tal y como se verá en los apartados siguientes.

4.2. Estratificación por plataforma

En esta sección, el conjunto de estudios va a dividirse en dos grandes subgrupos en función de la plataforma en la que el *malware* actúa: Windows o Android.

La información obtenida va a ser presentada en dos apartados, cada uno dedicado a mostrar los resultados de los experimentos realizados para cada medida. En cada uno de estos apartados se van a ofrecer los resultados de los metaanálisis por plataforma llevados a cabo. De este modo se facilitará la posibilidad de realizar comparativas entre los resultados obtenidos al realizar la estratificación, así como entre los resultados obtenidos por las distintas plataformas. Destacar también que queda fuera del ámbito de este trabajo la realización tanto de una metarregresión como de un análisis de subgrupos para intentar explicar la heterogeneidad, únicamente se va a realizar un metaanálisis independiente por cada tipo de plataforma en cada medida.

4.2.1. Metaanálisis de proporciones aplicado a la sensibilidad

Los dos experimentos realizados (Experimento 6 con datos de Windows y Experimento 7 con datos de Android) devuelven los siguientes resultados:

Windows:

```
    pred    ci.lb    ci.ub    pi.lb    pi.ub
0.977321 0.965787 0.986610 0.909959 1.000000

Random-Effects Model (k = 20; tau^2 estimator: DL)

tau^2 (estimated amount of total heterogeneity): 0.0057 (SE = 0.0049)
tau (square root of estimated tau^2 value):      0.0755
I^2 (total heterogeneity / total variability):   99.61%
H^2 (total variability / sampling variability):  254.15

```

```
Test for Heterogeneity:
Q(df = 19) = 4828.8005, p-val < .0001

Model Results:
estimate      se     zval    pval   ci.lb   ci.ub
  1.4166  0.0175  81.0125  <.0001  1.3823  1.4508  ***
---
Signif. codes:  0 "***" 0.001 "**" 0.01 "*" 0.05 "."
                0.1 " " 1

       estimate  ci.lb  ci.ub
tau^2      0.01   0.01   0.02
tau        0.08   0.07   0.14
I^2(%)    99.61  99.57  99.89
H^2      254.15 230.82 895.81
```

Android:

```
    pred    ci.lb    ci.ub    pi.lb    pi.ub
 0.980958 0.971623 0.988502 0.910521 1.000000

Random-Effects Model (k = 29; tau^2 estimator: DL)

tau^2 (estimated amount of total heterogeneity): 0.0068 (SE = 0.0049)
tau (square root of estimated tau^2 value):      0.0824
I^2 (total heterogeneity / total variability):   99.16%
H^2 (total variability / sampling variability):  119.65

Test for Heterogeneity:
Q(df = 28) = 3350.1005, p-val < .0001

Model Results:
estimate      se     zval    pval   ci.lb   ci.ub
  1.4307  0.0156  91.9134  <.0001  1.4002  1.4612  ***
---
Signif. codes:  0 "***" 0.001 "**" 0.01 "*" 0.05 "."
                0.1 " " 1

       estimate ci.lb  ci.ub
tau^2      0.01  0.00   0.01
tau        0.08  0.07   0.12
I^2(%)    99.16 98.73  99.57
H^2      119.65 78.85 233.28
```

Los efectos conjuntos estimados de la sensibilidad son 0,977321 para Windows y 0,980958 para Android con $IC(95\,\%) = (0{,}965787, 0{,}986610)$ e $IC(95\,\%) = (0{,}971623, 0{,}988502)$ respectivamente. Se observa que sus valores son bastante parecidos, rondando ambos el 98 % y siendo el intervalo de confianza de Android ligeramente más estrecho (apenas 4 milésimas menos) que el de Windows, lo que implica que su estimación es algo más confiable que la ofrecida por el metaanálisis de Windows, lo cual tiene sentido al disponer Windows de 20 estudios y Android de un total de 29. No es de extrañar, por tanto, que el valor obtenido en el Experimento 2 del metaanálisis de la sensibilidad global fuese 0,979532.

La varianza estimada de la heterogeneidad, τ^2, varía del 0,0057 de Windows al 0,0068 de Android, mientras que la que se obtuvo en el metaanálisis conjunto era algo menor, 0,0024. Por otro lado, el valor del test Q es de 4828,8005 y 3350,1005 para Windows y Android respectivamente ambos con $p < 0{,}0001$, mientras que en el caso conjunto era prácticamente el doble de la mayor 8196,8022. En este caso, el valor de I^2 es del 99,61 % en Windows y del 99,16 % en Android. Todos estos resultados indican que sigue existiendo heterogeneidad muy elevada en los metaanálisis realizados en ambas plataformas.

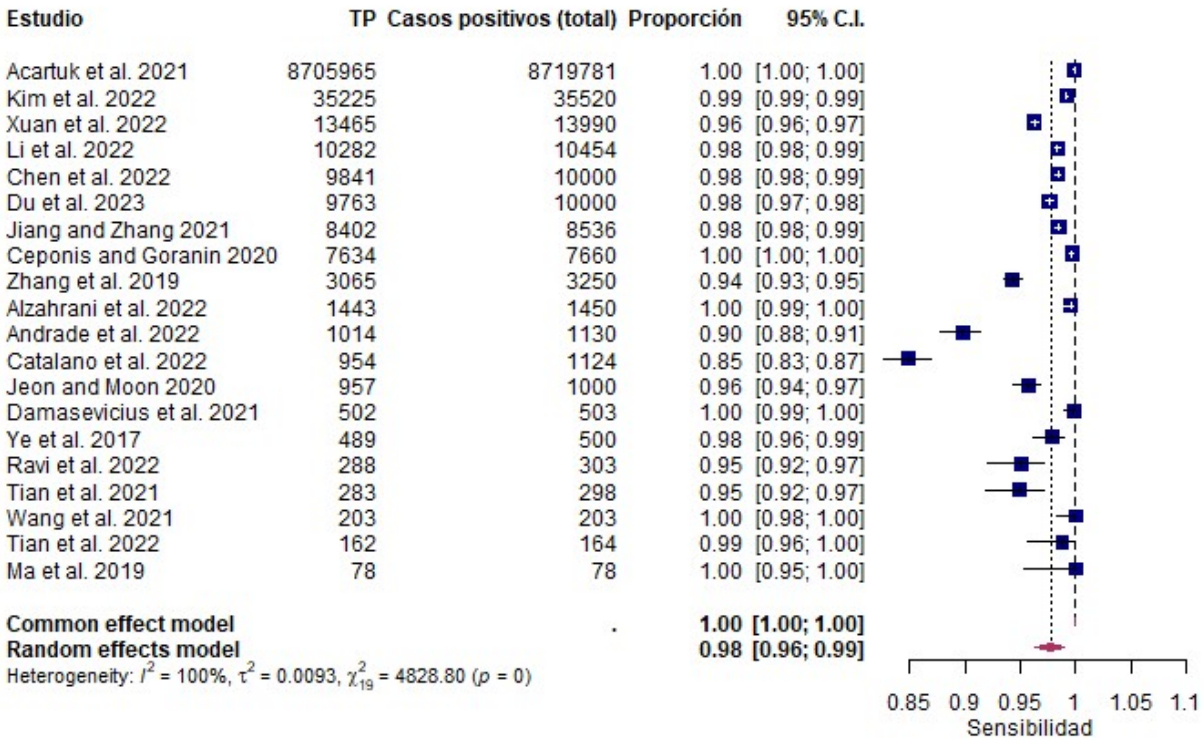

Figura 4.11: Experimento 6 - Diagrama de efectos sensibilidad en Windows

Una inspección visual sobre el diagrama de efectos de Windows que se encuentra en la Figura 4.11 indica que existen bastantes estudios que no se cruzan con la línea vertical del diamante, lo que confirma la alta heterogeneidad ofrecida por los test. Además, se podrían identificar como estudios sospechosos de ser atípicos a Catalano et al. [2022] y Andrade et al.

[2022]. Al ejecutar la función que calcula los estudios influyentes se observa que únicamente el primero de ellos, Catalano et al. [2022] es realmente influyente, sin embargo su eliminación no varía la existencia de heterogeneidad ya que el valor de I^2 sigue siendo superior al 99 %.

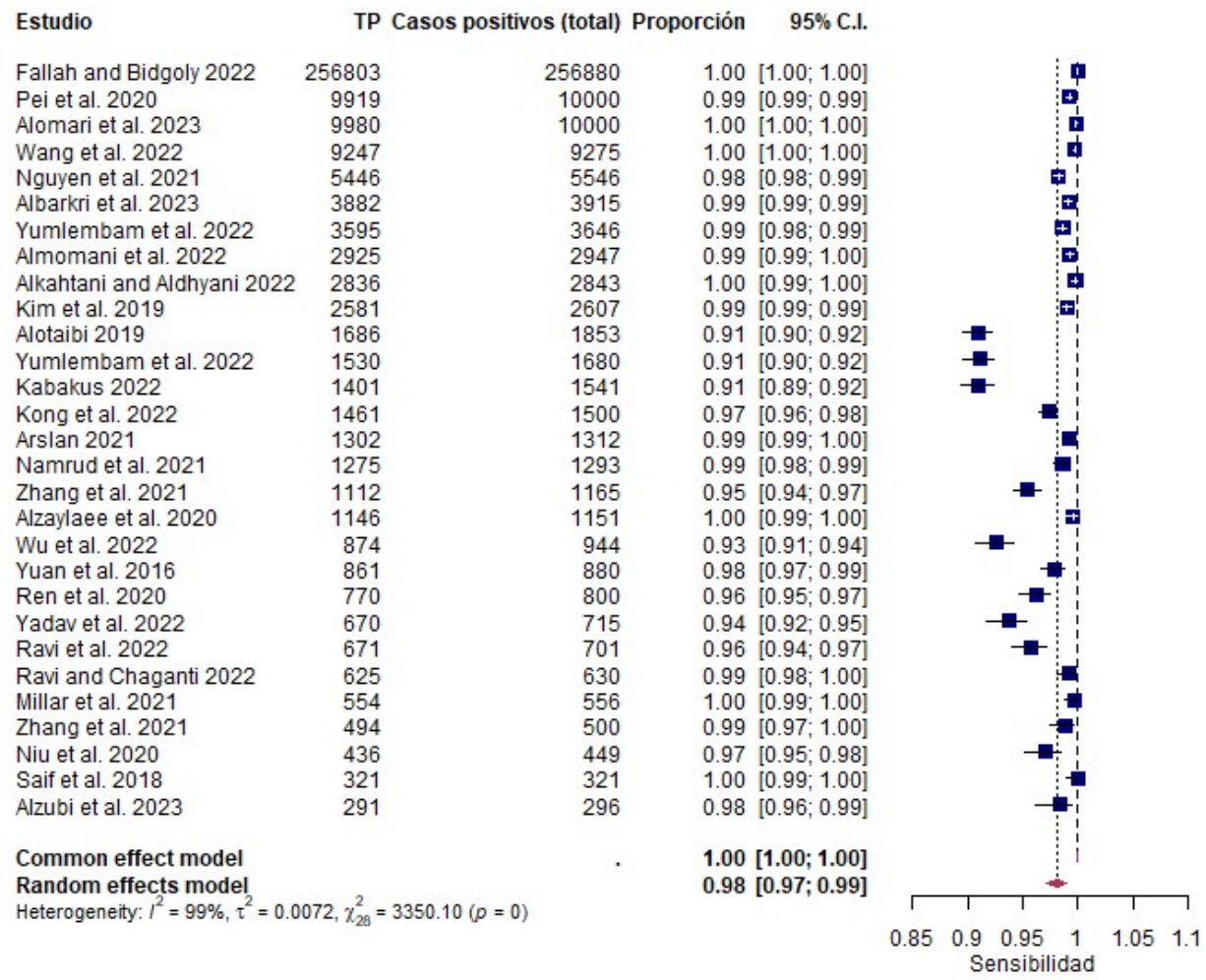

Figura 4.12: Experimento 7 - Diagrama de efectos sensibilidad en Android

Del mismo modo, al revisar el diagrama de efectos de Android de la Figura 4.12, se encuentran numerosos estudios alejados de la línea vertical del diamante, lo que explica la existencia de heterogeneidad. Aunque en esta ocasión los estudios no se encuentran tan alejados como en el caso de Windows, podrían ser sospechosos de ser atípicos Alotaibi [2019], Yumlembam et al. [2022] y Kabakus [2022], sin embargo, los resultados ofrecidos por el experimento respecto al cálculo de influyentes, indican que no existe ningún estudio influyente.

Adicionalmente, se han llevado a cabo algunos experimentos de combinaciones de estudios a eliminar en el caso de Windows. Sin embargo, incluso tras eliminar la mitad de estudios la heterogeneidad seguía siendo elevada. Para ver hasta que punto el valor de la heterogeneidad era inmutable se han seguido eliminando estudios y se ha constatado la heterogeneidad no ha pasado de ser considerable a ser significativa (bajando el valor de I^2 a entorno el 70 %)

hasta que no han quedado cuatro estudios de los veinte originales.

4.2.2. Metaanálisis de proporciones aplicado a la especificidad

Tras la realización de los dos experimentos experimentos (Experimento 8 con datos de Windows y Experimento 9 con datos de Android) se obtienen los siguientes resultados:

Windows:

```
    pred    ci.lb    ci.ub    pi.lb    pi.ub
0.972708 0.966361 0.978428 0.942592 0.992185

Random-Effects Model (k = 20; tau^2 estimator: DL)

tau^2 (estimated amount of total heterogeneity): 0.0014 (SE = 0.0013)
tau (square root of estimated tau^2 value):      0.0371
I^2 (total heterogeneity / total variability):   98.76%
H^2 (total variability / sampling variability):  80.61

Test for Heterogeneity:
Q(df = 19) = 1531.5176, p-val < .0001

Model Results:
estimate      se      zval    pval   ci.lb   ci.ub
  1.4020  0.0093  151.0865  <.0001  1.3838  1.4202  ***
---
Signif. codes:  0 "***" 0.001 "**" 0.01 "*" 0.05 "."
                0.1 " " 1

       estimate  ci.lb  ci.ub
tau^2      0.00   0.00   0.01
tau        0.04   0.05   0.11
I^2(%)    98.76  99.41  99.86
H^2       80.61 169.41 705.92
```

Android:

```
    pred    ci.lb    ci.ub    pi.lb    pi.ub
0.980160 0.973948 0.985565 0.939080 0.999253


Random-Effects Model (k = 29; tau^2 estimator: DL)

tau^2 (estimated amount of total heterogeneity): 0.0029 (SE = 0.0022)
```

```
tau (square root of estimated tau^2 value):      0.0536
I^2 (total heterogeneity / total variability):   98.71%
H^2 (total variability / sampling variability):  77.76

Test for Heterogeneity:
Q(df = 28) = 2177.3533, p-val < .0001

Model Results:
estimate      se      zval    pval   ci.lb   ci.ub
  1.4272  0.0105  136.5448  <.0001  1.4067  1.4476  ***
---
Signif. codes:  0 "***" 0.001 "**" 0.01 "*" 0.05 "."
                0.1 " " 1

       estimate  ci.lb  ci.ub
tau^2      0.00   0.00   0.01
tau        0.05   0.06   0.11
I^2(%)    98.71  99.07  99.70
H^2       77.76 107.69 328.34
```

La estimación del tamaño de efecto conjunto de la especificidad ofrece en Windows el valor 0,972708 y en Android el valor 0,980160 con unos intervalos de confianza al 95 % de (0,966361, 0,978428) y (0,973948, 0,985565) respectivamente. De nuevo se tienen valores bastante parejos que varían entre el 97 % y el 98 %. Mientras que en el metaanálisis global sobre la especificidad realizado en la Sección 4.1.2, el valor fue de 0,976382.

La varianza estimada de la heterogeneidad, τ^2, vuelve a ser superior en Android (0,0029) respecto a Windows (0,0014) del mismo modo que ocurría en la sensibilidad. Por su parte, en el metaanálisis conjunto era bastante superior a ambas (0,0058). Por otro lado, el test Q obtiene los valores de 1531,5176 y 2177,3533 en Windows y Android respectivamente ambos con $p < 0{,}0001$, mientras que en el caso conjunto era muy superior 20519,3192 siendo también significativo el resultado obtenido. Por último, el valor de I^2 es del 98,76 % en Windows y del 98,71 % en Android. Por lo que se puede concluir que la heterogeneidad elevada sigue existiendo tras haber realizado la división en subgrupos.

En el diagrama de efectos de Windows referente a la especificidad, que se encuentra en la Figura 4.13, se observa la presencia de heterogeneidad debido a la posición relativa de diversos estudios respecto de la línea vertical que representa el valor estimado conjunto de la especificidad. No obstante, se pueden identificar tres estudios que podrían ser atípicos: Damaševičius et al. [2021], Andrade et al. [2022] y Catalano et al. [2022] y se podría pensar que al eliminarlos se reduciría la heterogeneidad. El método de cálculo de los estudios influyentes de R confirma que los tres estudios son influyentes, sin embargo, su supresión no mejora los resultados obtenidos en los test de heterogeneidad, siendo el valor de I^2 cercano

al 99 %.

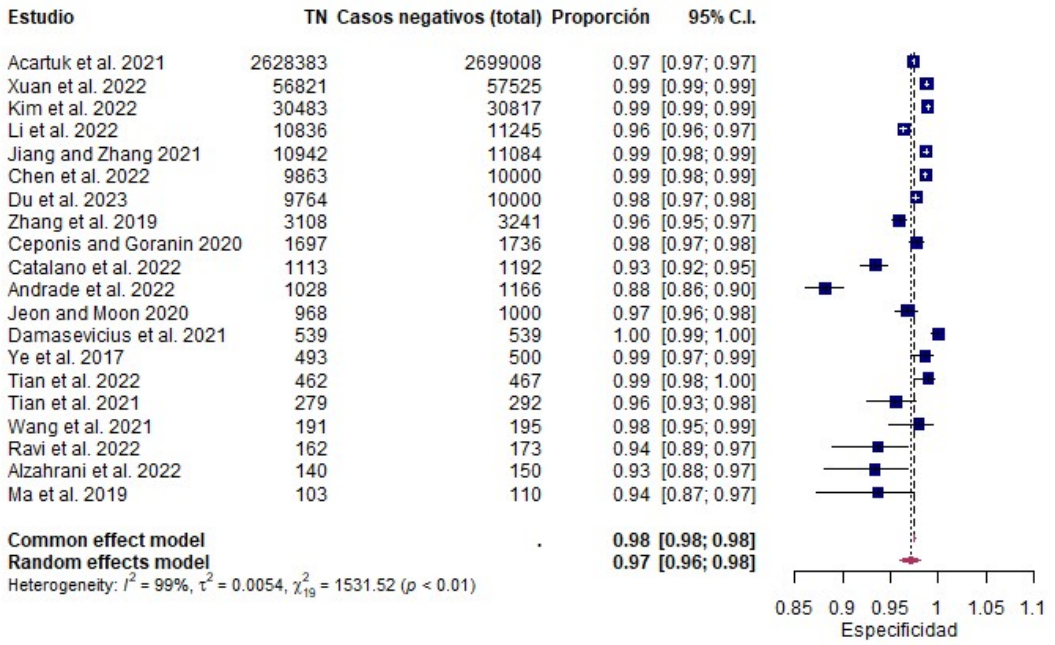

Figura 4.13: Experimento 8 - Diagrama de efectos especificidad en Windows

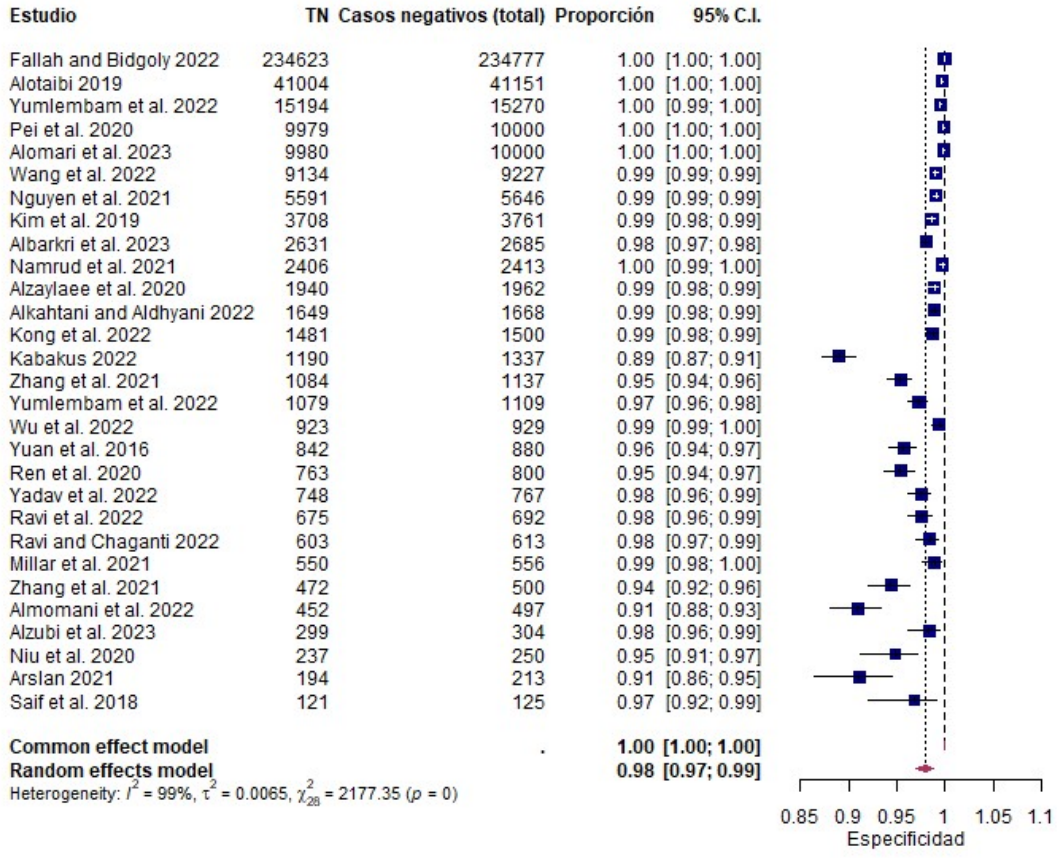

Figura 4.14: Experimento 9 - Diagrama de efectos especificidad en Android

Al revisar el diagrama de efectos de Android de la Figura 4.14, se encuentran numerosos estudios cuyo intervalo de confianza no toca al valor estimado, lo que explica la existencia de heterogeneidad. Podrían ser sospechosos de ser atípicos Kabakus [2022], Almomani et al. [2022] y Arslan [2021], sin embargo, los resultados ofrecidos por el experimento respecto al cálculo de influyentes, indican que solo los dos primeros lo son. Tras eliminarlos, la heterogeneidad se mantiene elevada, siendo $I^2 = 98,35\,\%$.

4.3. Estratificación por plataforma y tipo de análisis

En este último bloque de resultados, se va a añadir un segundo nivel de estratificación: el tipo de análisis. Como es conocido, el tipo de análisis en la detección de *malware* toma tres posibles valores: estático, dinámico o híbrido. De este modo, se va a proceder a realizar una estratificación en dos niveles, donde el primer nivel será la plataforma afectada y el segundo este tipo de análisis. Así, para cada una de las medidas se van a desarrollar cinco metaanálisis de forma independiente. Son únicamente cinco porque existe sólo un estudio de Android con análisis dinámico con lo que no es factible llevar a cabo un metaanálisis en ese subgrupo. La información obtenida va a ser presentada de nuevo en dos apartados, cada uno dedicado a una de las medidas que se están estudiando. Aunque en esta ocasión se van a omitir los resultados ofrecidos por R (salvo los diagramas de efectos) y en su lugar la información será expuesta en sendas tablas resumen.

4.3.1. Metaanálisis de proporciones aplicado a la sensibilidad

Se ejecutan 5 experimentos diferentes (Experimento 10 a Experimento 14) uno por cada uno de los metaanálisis que deben llevarse a cabo debido a la estratificación. En la Tabla 4.1 se muestra un resumen de los datos obtenidos en la predicción del efecto estimado y los distintos test de heterogeneidad para cada metaanálisis.

Tabla 4.1: Tabla resumen de datos de metaanálisis de sensibilidad estratificado (k = número de estudios)

Plataforma	**Análisis**	**k**	**Efecto estimado**	$\boldsymbol{\tau^2}$	**Q (p-valor)**	$\mathbf{I^2(\%)}$
Windows	Estático	10	0,971624	0,0111	867,1594($<$ 0,0001)	98,96
Windows	Dinámico	7	0,984190	0,0080	2246,6800($<$ 0,0001)	99,73
Windows	Híbrido	3	0,982807	0,0009	58,8316($<$ 0,0001)	96,60
Android	Estático	21	0,977288	0,0059	1131,2090($<$ 0,0001)	98,23
Android	Híbrido	7	0,986211	0,0060	246,1552($<$ 0,0001)	97,56

Se observa que el efecto estimado en cada uno de los subgrupos se mantiene entre el 97,1 % y el 98,7 %, no muy lejos entre sí ni de los valores calculados en los apartados anteriores. Lo

que si parece destacar es que tanto en Windows como en Android, el análisis estático es el que mayor número de estudios posee y al mismo tiempo el que peor rendimiento ofrece en cuanto a sensibilidad, aún siendo muy alto.

Por otro lado, los valores calculados de los diferentes test siguen indicando una heterogeneidad muy alta. Dentro de ellos parece que el caso híbrido es el que se comporta ligeramente mejor pero el menor número de estudios comparativamente con los otros casos puede afectar significativamente en este caso.

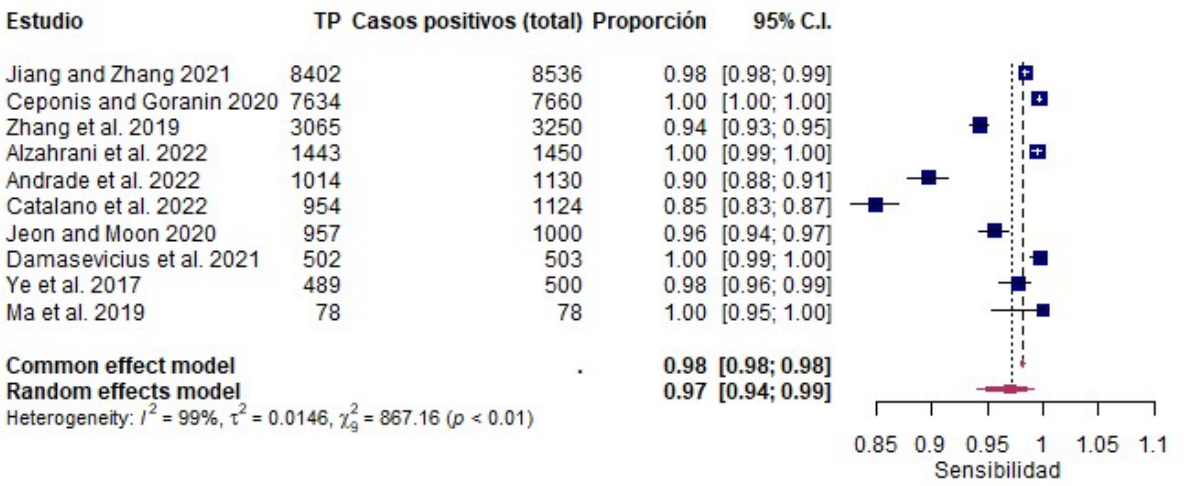

Figura 4.15: Experimento 10 - Diagrama de efectos sensibilidad en Windows Estático

En la Figura 4.15 que muestra el diagrama de efectos del metaanálisis de sensibilidad aplicado al subgrupo de Windows con análisis estático, se observa la presencia de heterogeneidad y de dos posibles estudios atípicos: Catalano et al. [2022] y Andrade et al. [2022]. Esto se confirma a través del cálculo de influyentes en dos pasos, por lo que se excluyen ambos estudios. Sin embargo su exclusión no evita la presencia de una alta heterogeneidad con valores de I^2 por encima del 98 %.

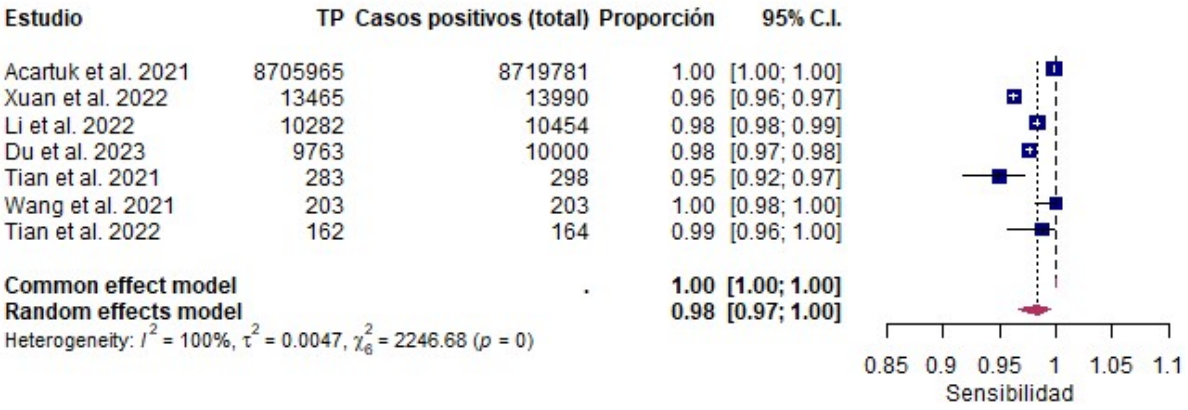

Figura 4.16: Experimento 11 - Diagrama de efectos sensibilidad en Windows Dinámico

En la Figura 4.16 que muestra el diagrama de efectos del metaanálisis de sensibilidad aplicado al subgrupo de Windows con análisis dinámico, también se observa cierta heterogeneidad. A simple vista no destaca ningún estudio por estar especialmente alejado del valor estimado por lo que parecería que no existen valores atípicos. Sin embargo, tras realizar el cálculo de influyentes en varios pasos aparecen estudios que sí lo son: Acarturk et al. [2021] en primer lugar, Wang et al. [2022] y Tian et al. [2021] en segundo y Xuan et al. [2022] en tercero. Al realizar la última exclusión la heterogeneidad sigue siendo alta pero el valor de I^2 se reduce hasta el $85,67\,\%$. Aunque en el último paso siguen apareciendo dos de los tres estudios como influyentes (los dos de mayor tamaño), se decide no aplicar la eliminación ya que en dicho caso solo quedaría un estudio y no tendría sentido.

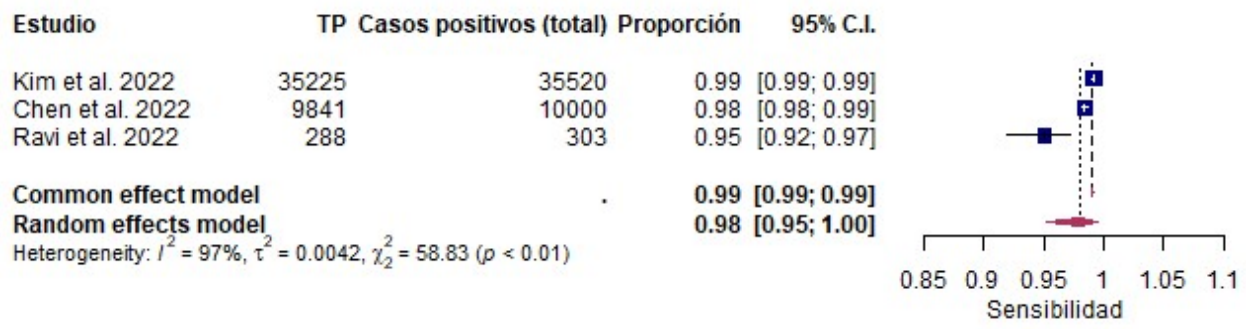

Figura 4.17: Experimento 12 - Diagrama de efectos sensibilidad en Windows Híbrido

Por su parte, en la Figura 4.17 se presenta el diagrama de efectos del metaanálisis de sensibilidad aplicado al subgrupo de Windows con análisis híbrido. Se observa que solo posee tres estudios, que dos de ellos son muy precisos debido a la cantidad de datos que poseen y el tercero con muchos menos datos en relación a los anteriores, tienen un intervalo de confianza más amplio. Esta situación provoca heterogeneidad. Se podría decir que el tercer estudio es atípico, pero los cálculos de influyentes indican que los tres lo son. No obstante, si se elimina el tercer estudio, la heterogeneidad pasa a ser no relevante con un valor de I^2 de 38,87.

Al analizar la Figura 4.18, se observa que el diagrama de efectos de sensibilidad en Android aplicando análisis estático ofrece heterogeneidad. En este caso, similar a otros anteriores, los estudios con mayor peso, es decir, los que tienen un mayor número de muestras, tienden a estar a la derecha de la sensibilidad estimada conjunta, mientras que estudios de tamaño intermedio tienden a estar en el lado contrario, provocando así la heterogeneidad. Por otro lado, no parece que existan estudios atípicos, y así lo confirma el cálculo de influyentes.

En el último de los diagramas de efecto de sensibilidad, aplicado en este caso a Android con análisis híbrido que aparece en la Figura 4.19, se puede comprobar también la presencia de heterogeneidad, donde los estudios más influyentes se encuentran a la derecha y los menos influyentes (a excepción del último) a la izquierda. Viendo el diagrama, el estudio Wu et al. [2022] sería candidato a ser atípico y así lo confirma el cálculo de influyentes. No obstante, también incluye los artículos Yuan et al. [2016] y Ravi et al. [2022]. La supresión de estos

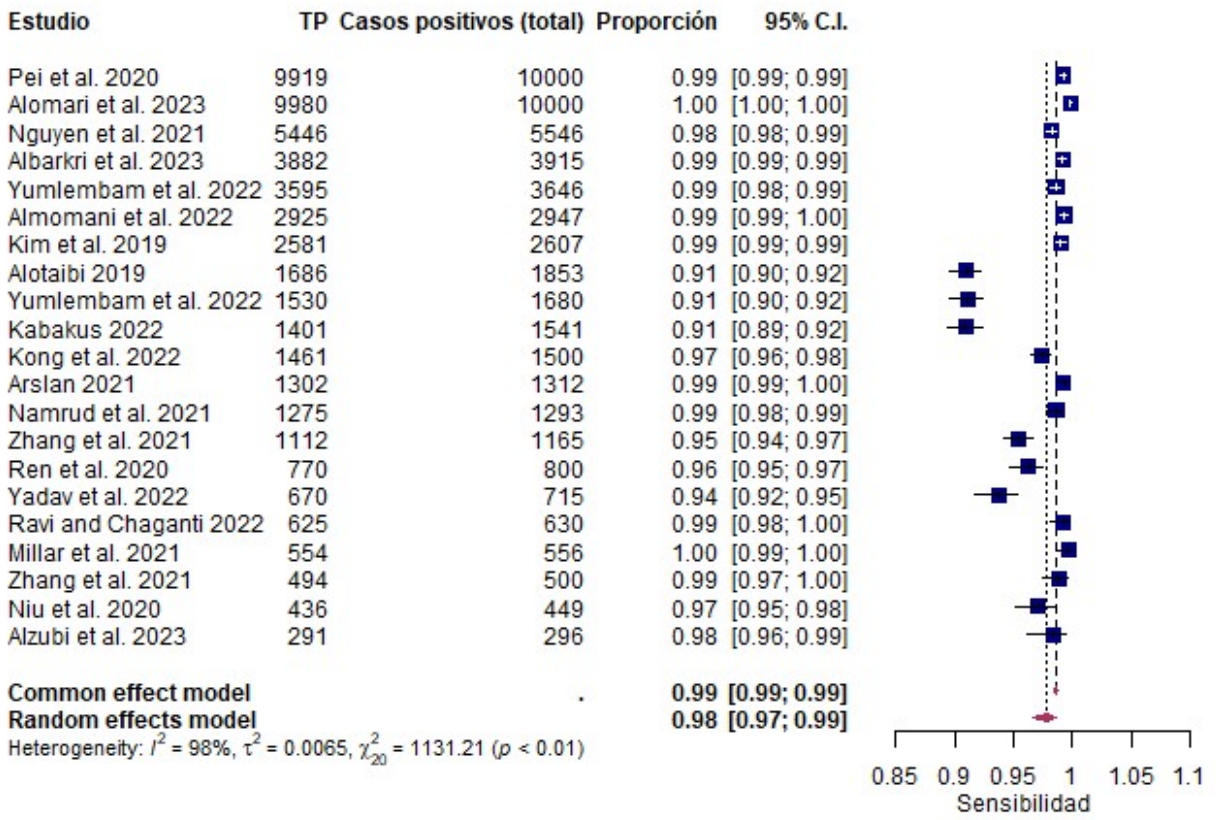

Figura 4.18: Experimento 13 - Diagrama de efectos sensibilidad en Android Estático

tres artículos, que son precisamente los que se encontraban a la izquierda del valor estimado, provoca que no exista heterogeneidad alguna ya que el valor estimado de la sensibilidad es 1 al igual que el resto de los estudios.

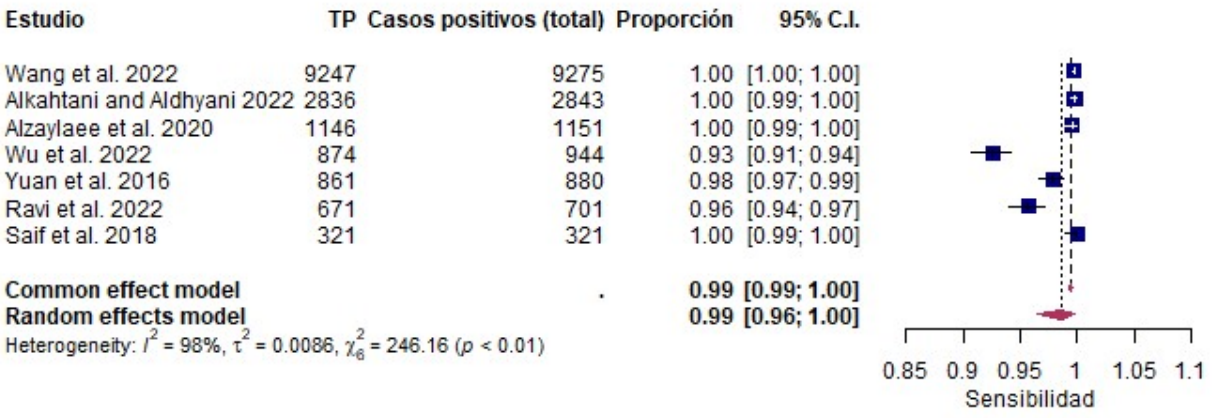

Figura 4.19: Experimento 14 - Diagrama de efectos sensibilidad en Android Híbrido

4.3.2. Metaanálisis de proporciones aplicado a la especificidad

Se proponen otros 5 experimentos en R (Experimento 15 a Experimento 19) por cada uno de los metaanálisis que deben llevarse a cabo debido a la estratificación. En la Tabla 4.2 se muestra un resumen de los datos obtenidos tanto en la predicción del efecto estimado conjunto de la especificidad como en los distintos test de heterogeneidad para cada meta-análisis.

Tabla 4.2: Tabla resumen de datos de metaanálisis de especificidad estratificado (k = número de estudios)

Plataforma	Análisis	k	Efecto estimado	τ^2	Q (p-valor)	I^2(%)
Windows	Estático	10	0,964855	0,0070	397,6923(< 0,0001)	97,74
Windows	Dinámico	7	0,976864	0,0010	652,3381(< 0,0001)	99,08
Windows	Híbrido	3	0,985744	0,0003	21,2645(< 0,0001)	90,59
Android	Estático	21	0,977186	0,0033	1044,1204(< 0,0001)	98,08
Android	Híbrido	7	0,983876	0,0012	51,3509(< 0,0001)	88,32

Se observa que el efecto estimado en cada uno de los subgrupos se varía entre el 96, 4 % y el 98, 6 %, no muy lejos entre sí ni de los valores calculados en los apartados anteriores. Tal y como ocurría con la sensibilidad, tanto en Windows como en Android, el análisis estático es el que peor rendimiento parece ofrecer al evaluar la especificidad estimada, aún siendo su valor muy alto. Por su parte, el análisis híbrido destaca de forma positiva en este aspecto. En cuanto a los resultados ofrecidos por los test, siguen demostrando una heterogeneidad muy elevada. No obstante, en el caso del análisis híbrido se perciben unos valores de I^2 muy inferiores al resto tanto en Windows como en Android.

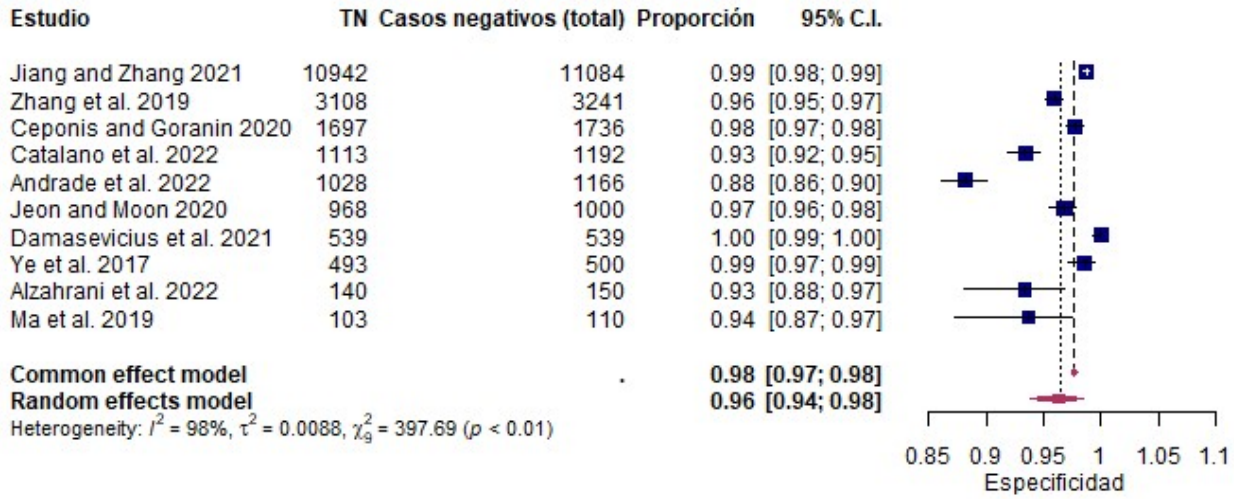

Figura 4.20: Experimento 15 - Diagrama de efectos especificidad en Windows Estático

En la Figura 4.20, que muestra el diagrama de efectos del metaanálisis de especificidad

aplicado al subgrupo de Windows con análisis estático, se observa la presencia de heterogeneidad y de al menos un posible estudio atípico: Andrade et al. [2022]. Mediante el cálculo de influyentes se confirma que este estudio lo es, así como también Damaševičius et al. [2021]. Sin embargo su exclusión no evita la presencia de una alta heterogeneidad con valores de I^2 cercanos del 98 %.

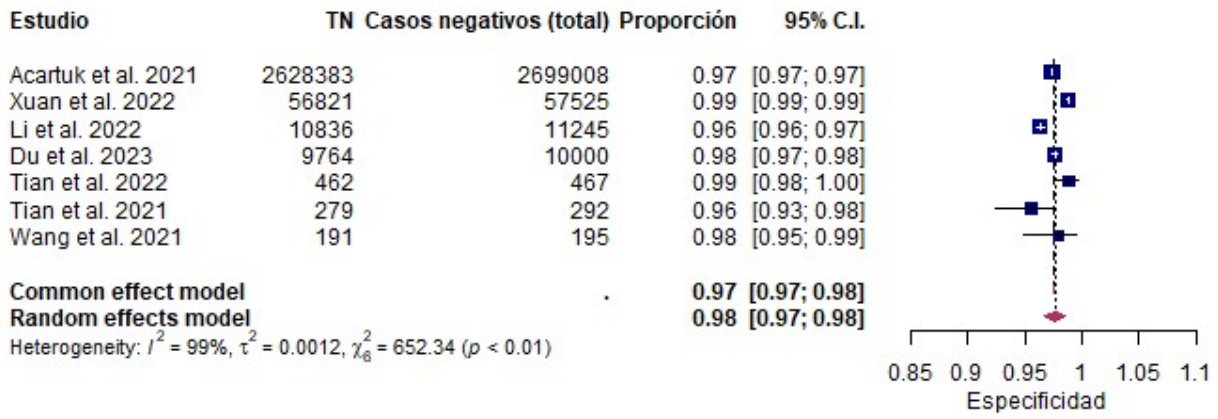

Figura 4.21: Experimento 16 - Diagrama de efectos especificidad en Windows Dinámico

En la Figura 4.21, que muestra el diagrama de efectos del metaanálisis de especificidad aplicado al subgrupo de Windows con análisis dinámico, también se observa cierta heterogeneidad. Aunque en esta ocasión el estudio con mayor peso cae justo sobre el valor estimado y el resto de estudios no se alejan en exceso, lo que hace pensar que no existen valores atípicos. Sin embargo, tras realizar el cálculo de influyentes en varios pasos aparecen estudios que sí lo son: Xuan et al. [2022] en primer lugar, y Li et al. [2022] en segundo. Al realizar la exclusión de ambos estudios la heterogeneidad se reduce, aunque sigue siendo significativa según el valor de I^2 (65,30 %).

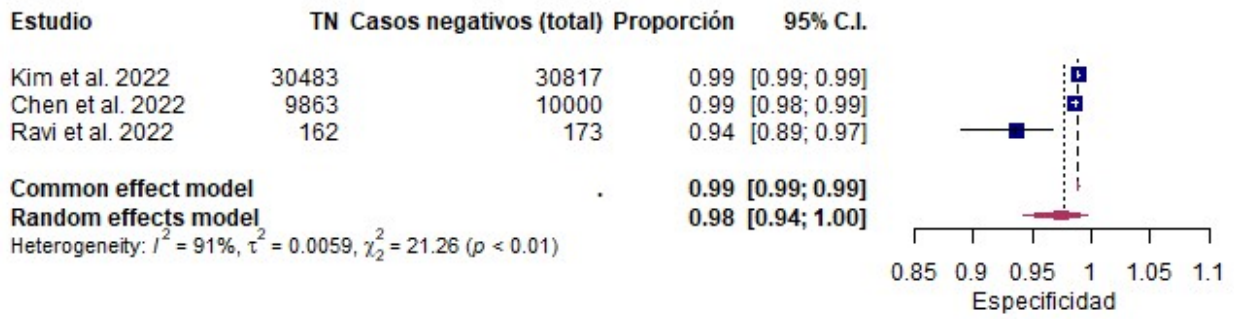

Figura 4.22: Experimento 17 - Diagrama de efectos especificidad en Windows Híbrido

Por su parte, en la Figura 4.22 se presenta el diagrama de efectos del metaanálisis de especificidad aplicado al subgrupo de Windows con análisis híbrido. Al igual que ocurría con la sensibilidad, se observa heterogeneidad debido a que dos sus estudios son muy precisos

por la cantidad de datos que poseen, y que el tercero, con muchos menos datos, tiene un intervalo de confianza más amplio. No se destacaría ningún estudio como atípico al revisar el diagrama, pero los cálculos de influyentes indican que lo son los tres.

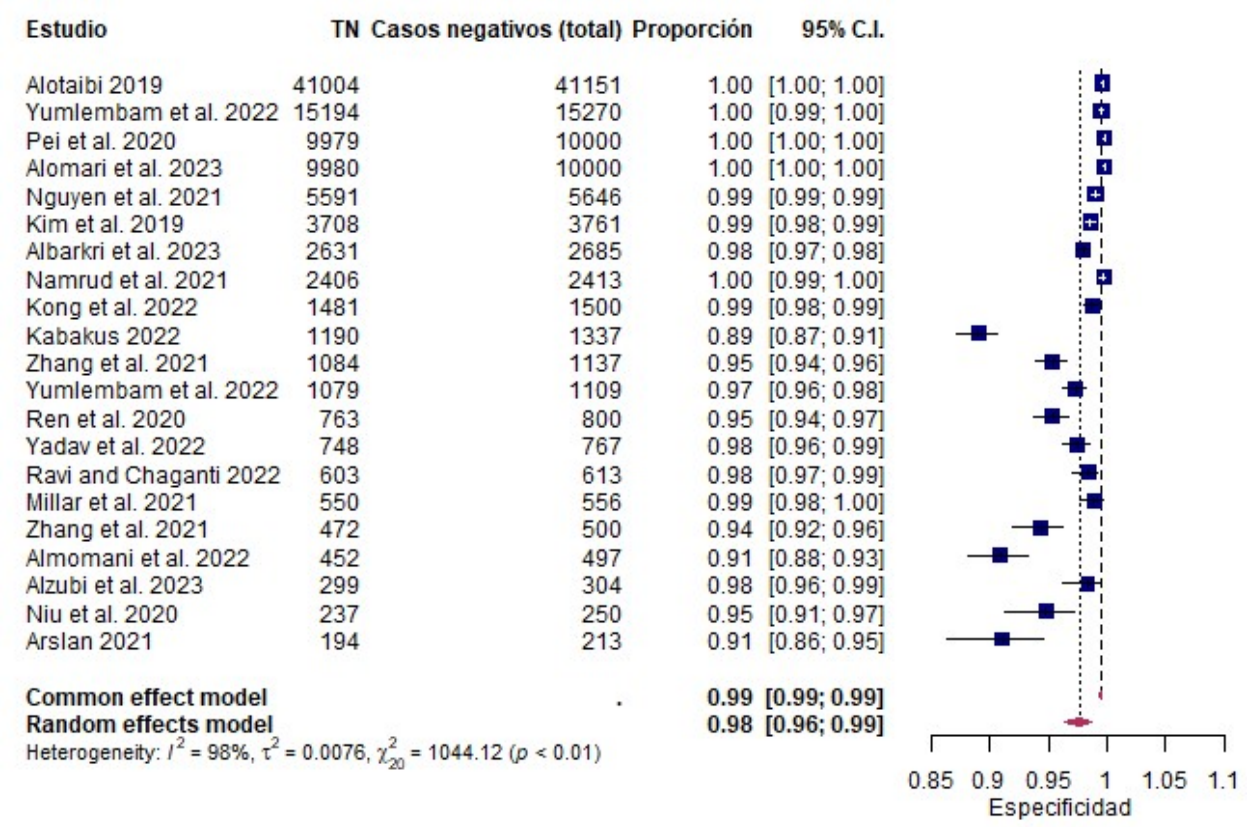

Figura 4.23: Experimento 18 - Diagrama de efectos especificidad en Android Estático

Al analizar la Figura 4.23, se observa que el diagrama de efectos de especificidad en Android aplicando análisis estático ofrece heterogeneidad. De nuevo, se da la situación de que los estudios con mayor peso tienden a estar a la derecha de la especificidad estimada conjunta, mientras que estudios de tamaño intermedio tienden a estar en el lado contrario, provocando así la heterogeneidad. Por otro lado, el estudio Kabakus [2022] es candidato a ser atípico. Al realizar el cálculo de influyentes no solo se observa que el estudio anterior lo es, si no también Almomani et al. [2022]. Sin embargo, no cambia el estado de la heterogeneidad cuando se suprimen dichos estudios.

En el último de los diagramas de efecto de especificidad, aplicado en este caso a Android con análisis híbrido que aparece en la Figura 4.24, se puede comprobar también la presencia de heterogeneidad pero menos pronunciada que en los casos anteriores. Viendo el diagrama, el estudio Yuan et al. [2016] sería candidato a ser atípico y así lo confirma el cálculo de influyentes. No obstante, también incluye los artículos Wang et al. [2022] y Ravi et al. [2022]. La supresión del primero y el tercero de estos artículos, que son precisamente los que se encontraban a la izquierda del valor estimado, provoca que la heterogeneidad baje

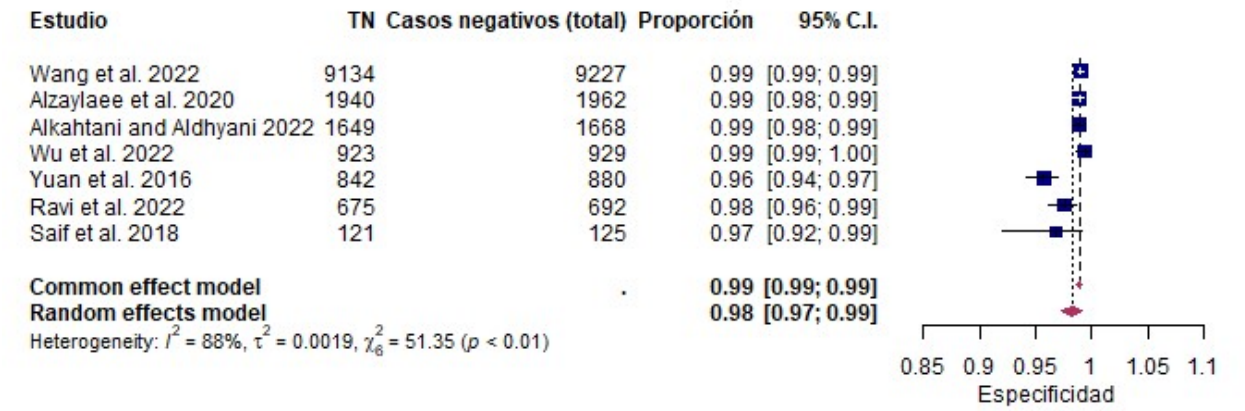

Figura 4.24: Experimento 19 - Diagrama de efectos especificidad en Android Híbrido

drásticamente, pasando a ser irrelevante ($I^2 = 29{,}26\,\%$).

Capítulo 5

Discusión

En este capítulo se discutirán los resultados obtenidos por los distintos experimentos en el apartado anterior, indicando los posibles problemas que pueden surgir al tomar ciertas decisiones y ofreciendo una conclusión final al respecto de los resultados ofrecidos por los metaanálisis realizados. La meta final será dar respuesta a los objetivos que se marcaron en la Sección 1.2 de este trabajo.

Antes de comenzar a analizar los resultados de los experimentos, se van a recordar algunas de las características de los estudios usados en los metaanálisis y que pueden influir en el análisis posterior:

- La búsqueda y filtrado de los estudios relacionados con la detección de malware mediante la aplicación de técnicas de aprendizaje automático dio como lugar a la obtención de 47 estudios para su inclusión en el metaanálisis (19 exclusivos de Windows, 27 exclusivos de Android y 1 que se aplica a ambas plataformas). Además, uno de los estudios de Android analiza dos conjuntos de datos, por lo que en total el metaanálisis dispone de 49 entradas distintas.

- Los valores de sensibilidad de estos estudios varían entre un 84,88 % y un 100 %, siendo solo 9 de ellos menores que el 95 %, por lo que se puede estimar que poseen en general una alta sensibilidad.

- Por otro lado, los valores de especificidad van desde el 88,16 % hasta el 100 %, con solo 10 de ellos menores al 95 %, de modo que también poseen en general una alta especificidad.

- Hay bastante variedad en los tamaños muestrales usados en los estudios (casos positivos en el caso de la sensibilidad y casos negativos en el caso de la especificidad). Existen estudios de menos de 200 muestras y también los hay de más de 10.000, incluso el más grande supera ampliamente el millón. Aunque la mayor parte se encuentra entre los 500 y los 10.000.

La evaluación de los distintos experimentos ha arrojado resultados interesantes pero al mismo tiempo han quedado algunos desafíos que requerirán de una consideración más detallada:

- **Heterogeneidad**. Los distintos metaanálisis realizados, incluidos aquellos en los que se han excluido ciertos artículos muestran de forma consistente, y salvo casos excepcionales, una muy alta heterogeneidad. Se podría pensar que esta heterogeneidad es debida a algunos factores no identificados que estuvieran influyendo en la efectividad de la detección, sin embargo, incluso en los casos en que existen pocos estudios y poseen unos valores de sensibilidad o especificidad cercanos, la heterogeneidad sigue presente.

  Una posible causa que genere que los resultados de los test de heterogeneidad, y en particular el valor de I^2, sean tan altos, se puede encontrar analizando los distintos diagramas de efectos que se presentaron en el Capítulo 4 de resultados. El intervalo de confianza generado por cada estudio es dependiente del número de muestras que posee cada estudio. Así, estudios con pocas muestras tienen intervalos de confianza mayores mientras que los que poseen un gran número de muestras tienen un intervalo de confianza minúsculo. Esto provoca que, cuando varios estudios grandes o muy grandes difieren ligeramente en su proporción (del orden de la centésima o menos), no colisionen contra la estimación conjunta de la proporción, elevando así el resultado de I^2, y por tanto la heterogeneidad del metaanálisis. Además, en los resultados de este trabajo se observa que la mayor parte de los estudios pequeños y medianos tienden a tener una proporción bastante menor que los estudios grandes (lo cual tiene sentido ya que parten de un menor conjunto de entrenamiento y por tanto peores métricas) y, aún con un intervalo de confianza amplio, no llegan a contactar con la estimación conjunta de la proporción, elevando de nuevo la heterogeneidad.

  Dicho lo cual, aunque los estudios de pequeño tamaño pueden influir en la aparición de heterogeneidad, es la presencia de estudios muy grandes que varían ligeramente en su proporción la que más puede llegar a afectar en la medida de la heterogeneidad realizada por estos test. Aquí entraría, por tanto, recordar lo mencionado en la Sección 2.1.7 acerca de los test de heterogeneidad para metaanálisis de proporciones. Barker et al. [2021] indica que no hay test específicos para evaluar la heterogeneidad en este tipo de metaanálisis y que, aunque se puedan usar los test clásicos de metaanálisis tradicional, dado que la varianza de los datos es pequeña, el valor de I^2 suele ser alto. Así, es esperable que, en general, dicho valor sea elevado y que exista cierta heterogeneidad aunque esto no tiene porque derivar necesariamente en inconsistencia entre los estudios.

  Asumiendo la presencia de heterogeneidad, durante la fase de experimentación se ha realizado la exclusión de algunos estudios que podrían ser considerados atípicos. Sin embargo, en la mayoría de los casos esta exclusión no ha reducido la cantidad de heterogeneidad y cuando lo ha hecho ha sido a costa de dejar un número muy reducido

de estudios con lo que esta exclusión no sería del todo correcta. No obstante, se han analizado al detalle los estudios propuestos para descarte y no se detecta una razón de peso que pueda dar lugar a suprimir dichos estudios, ya que en caso de eliminación de estudios que no deberían ser excluidos el sesgo puede aumentar.

- **Sesgo de publicaciones**. En el capítulo de resultados ya se mencionó que no se iban a mostrar los resultados obtenidos por el diagrama del embudo, a excepción del primero que servía como ejemplo de lo que ocurría en el resto de metaanálisis realizados. Esta decisión está apoyada en el artículo de Barker et al. [2021] donde también indica que ni los test de Egger y Begg ni el diagrama de embudo son adecuados para usarse en los metaanálisis de proporciones ya que, por la forma en la que se definen tanto los test como el diagrama, se asume la hipótesis de la existencia de más resultados favorables que negativos. Esto tiene sentido en los metaanálisis tradicionales, pero no en los de proporciones donde no se definen estos conceptos. Así, lo que se aconseja es llevar a cabo una evaluación cualitativa de estudios como alternativa pero dicho estudio se sale del ámbito de este trabajo.

Teniendo en cuenta todas las consideraciones anteriores, y por mantener la cautela respecto a los datos obtenidos, se va a considerar que, efectivamente, existe heterogeneidad por lo que no se debería usar la estimación conjunta obtenida. No obstante, tras la realización de un análisis exhaustivo de los estudios, y visto que la mayor parte de los estudios ofrecían una sensibilidad y una especificidad alta, sobre todo aquellos con mayor número de muestras, se puede concluir que la aplicación de técnicas de aprendizaje profundo son muy útiles para la detección correcta de malware.

Por otro lado, los metaanálisis agrupados tanto por plataforma como por plataforma y tipo de análisis demuestran buenos niveles de detección, siendo el tipo de análisis híbrido el que mejores resultados ofrece en ambas métricas.

Por último, tras la exploración de los estudios y el análisis de los datos, no se observa ningún modelo que destaque en rendimiento respecto de otro. No obstante, el tipo de modelo más usado son las redes convolucionales o alguna de sus variantes (29 veces) ya sea solas o acompañadas por otra red, seguidas de las LSTM y sus variantes (14 apariciones).

Capítulo 6

Conclusiones y trabajos futuros

Este trabajo ha proporcionado una visión profunda sobre la aplicación de *deep learning* en la detección de *malware*. Mediante la revisión sistemática, la síntesis y la aplicación de metaanálisis se ha podido confirmar que el aprendizaje profundo es una herramienta poderosa y bastante efectiva para abordar los desafíos actuales y futuros que se puedan presentar en este ámbito. Aunque la heterogeneidad ofrecida por los estudios no permite ofrecer una medida de efecto final que resuma las métricas escogidas, entre otras razones por las propias dificultades para medir la heterogeneidad de forma adecuada en un metaanálisis de proporciones, se ha podido comprobar que el conjunto de valores en el que se mueven estas medidas está rondando o superando el 90 % en su mayoría, por lo que se puede afirmar sin temor que se trata de una herramienta valiosa para combatir el *malware*.

Aunque el análisis que se ha realizado en este trabajo ha pretendido indagar en distintas características que pueden influir en la aplicación de la detección de *malware* mediante herramientas de aprendizaje profundo, no se ha podido ahondar en algunas cuestiones que podrían ser también interesantes como son: el tipo de modelo usado, los conjuntos de datos usados, o la forma en la que los distintos estudios extraen las características del *malware* que posteriormente se incluyen en estos conjuntos de datos. Por otro lado, los estudios que se han analizado en este trabajo no han tratado ataques complejos que suelen escaparse a la detección. Tampoco se ha podido analizar como funcionan las herramientas de aprendizaje profundo en el ámbito de ciertos tipos de *malware* concretos ni se ha evaluado si estos modelos funcionan tan bien en el ámbito de la clasificación de los distintos tipos o familias de *malware*. Otra cuestión que se ha escapado a los límites de este trabajo es la realización de un análisis del sesgo de publicación de los estudios llevando a cabo una profunda evaluación cualitativa como alternativa al fallido intento de análisis a través de las herramientas explicadas para su detección.

Estas y otras limitaciones que posee este trabajo permiten a su vez abrir nuevas vías de investigación que podrían ser de aplicación en el futuro:

- Una primera cuestión que podría ampliar el análisis llevado a cabo consistiría en evaluar

si las posibles causas de la heterogeneidad tienen que ver con los niveles de estratificación que se han usado. Para ello convendría llevar a cabo una metarregresión, en particular un análisis por subgrupos tomando como variable moderadora el tipo de análisis o la plataforma.

- En segundo lugar, se podría realizar un estudio por otras variables no consideradas en los metaanálisis, como puede ser el tipo de modelo de aprendizaje profundo usado, el conjunto de datos usado como malware en el estudio, el año o los procesos de extracción de características, entre otras cuestiones.

- Basado en los mismos estudios e incorporando futuros, siempre que dispongan de la matriz de confusión, se podrían realizar otros metaanálisis que analizasen otras métricas como pueden ser la exactitud, la precisión o la medida F1.

- Otra posibilidad podría ser realizar un metaanálisis tomando la especificidad y la sensibilidad conjuntamente en lugar de por separado como se ha hecho en este trabajo.

- También podría ampliarse este estudio realizando un metaanálisis tradicional sobre alguna variable que lo permita, como el *odds ratio*, de modo que pudiera comprobarse si para esa medida siguen apareciendo esos valores de heterogeneidad tan altos. Y, al ser un metaanálisis tradicional, se podría evaluar de forma adecuada el sesgo con diagramas de embudo y otras herramientas.

- Otra cuestión interesante podría ser evaluar únicamente los estudios que tratan *malware* con ataques más complejos de defender como son los ataques adversarios, los de día cero o los metamórficos.

- También se podría realizar un estudio de la aplicación del aprendizaje profundo en la clasificación de *malware*, ya sea de forma general como se ha llevado a cabo en este caso, o bien centrándose en las familias existentes para *malware* específico.

- Siguiendo con la ciberseguridad, otro tema interesante podría ser centrarse en otro tipo de ataques no *malware* como pueden ser el *phishing* o los ataques *DDos*.

Por último, dada la inexistencia de metaanálisis en este ámbito y que se trata de un tema en auge que acaba de despegar hace pocos años, destacar que existe la intención de presentar este trabajo como publicación en alguna revista científica de impacto.

Bibliografía

International Telecommunication Union (ITU). Facts and Figures 2022 - Internet use, 2023. URL https://www.itu.int/itu-d/reports/statistics/2022/11/24/ff22-internet-use/.

AAG IT. AAG Latest Cyber-crime statistics, 2023. URL https://aag-it.com/the-latest-cyber-crime-statistics/.

AV-Test Institute. AVTest Malware statistics, 2023. URL https://www.av-test.org/en/statistics/malware.

Mihalj Bakator and Dragica Radosav. Deep Learning and Medical Diagnosis: A Review of Literature. *Multimodal Technologies and Interaction*, 2(3):47, September 2018. ISSN 2414-4088. doi: 10.3390/mti2030047. URL https://www.mdpi.com/2414-4088/2/3/47.

Khan Muhammad, Amin Ullah, Jaime Lloret, Javier Del Ser, and Victor Hugo C. de Albuquerque. Deep Learning for Safe Autonomous Driving: Current Challenges and Future Directions. *IEEE Transactions on Intelligent Transportation Systems*, 22(7):4316–4336, July 2021. ISSN 1558-0016. doi: 10.1109/TITS.2020.3032227. URL https://ieeexplore.ieee.org/abstract/document/9284628.

Weiwei Jiang. Applications of deep learning in stock market prediction: Recent progress. *Expert Systems with Applications*, 184:115537, December 2021. ISSN 0957-4174. doi: 10.1016/j.eswa.2021.115537. URL https://www.sciencedirect.com/science/article/pii/S0957417421009441.

Neelam Chandolikar, Chaitanya Joshi, Prateek Roy, Abhijeet Gawas, and Mini Vishwakarma. Voice Recognition: A Comprehensive Survey. In *2022 International Mobile and Embedded Technology Conference (MECON)*, pages 45–51, March 2022. doi: 10.1109/MECON53876.2022.9751903. URL https://ieeexplore.ieee.org/abstract/document/9751903.

Daniel W. Otter, Julian R. Medina, and Jugal K. Kalita. A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):604–624, February 2021. ISSN 2162-2388. doi: 10.1109/TNNLS.2020.2979670. URL https://ieeexplore.ieee.org/abstract/document/9075398.

Xiaodong He and Li Deng. Deep Learning for Image-to-Text Generation: A Technical Overview. *IEEE Signal Processing Magazine*, 34(6):109–116, November 2017. ISSN 1558-0792. doi: 10.1109/MSP.2017.2741510. URL https://ieeexplore.ieee.org/abstract/document/8103169.

Daniel Berman, Anna Buczak, Jeffrey Chavis, and Cherita Corbett. A Survey of Deep Learning Methods for Cyber Security. *Information*, 10(4):122, April 2019. ISSN 2078-2489. doi: 10.3390/info10040122. URL https://www.mdpi.com/2078-2489/10/4/122.

Robert W. Palmatier, Mark B. Houston, and John Hulland. Review articles: purpose, process, and structure. *Journal of the Academy of Marketing Science*, 46(1):1–5, January 2018. ISSN 0092-0703, 1552-7824. doi: 10.1007/s11747-017-0563-4. URL http://link.springer.com/10.1007/s11747-017-0563-4.

R. Aguilera Eguía. ¿Revisión sistemática, revisión narrativa o metaanálisis? *Revista de la Sociedad Española del Dolor*, 21(6):359–360, December 2014. ISSN 1134-8046. doi: 10.4321/S1134-80462014000600010. URL https://scielo.isciii.es/scielo.php?script=sci_abstract&pid=S1134-80462014000600010&lng=es&nrm=iso&tlng=es.

Vicente Javier Escrig Sos, José Antonio Llueca Abella, Laura Granel Villach, and Manuel Bellver Oliver. Metaanálisis: una forma básica de entender e interpretar su evidencia. *Revista de Senología y Patología Mamaria*, 34(1):44–51, January 2021. ISSN 0214-1582. doi: 10.1016/j.senol.2020.05.007. URL https://www.sciencedirect.com/science/article/pii/S0214158220300700.

Michael Borenstein, Larry V. Hedges, Julian P. T. Higgins, and Hannah R. Rothstein. *Introduction to Meta-Analysis*. Wiley, 1 edition, March 2009. ISBN 9780470057247 9780470743386. doi: 10.1002/9780470743386. URL https://onlinelibrary.wiley.com/doi/book/10.1002/9780470743386.

Steven Higgins. *A Brief History of Meta-analysis*. Cambridge University Press, Cambridge, 2018. ISBN 9781107033320. doi: 10.1017/9781139519618.002. URL https://www.cambridge.org/core/books/improving-learning/brief-history-of-metaanalysis/29CF56ABBF7BD3939C6F0764095540C4.

Mathias Harrer, Pim Cuijpers, Toshi Furukawa, and David Ebert. *Doing meta-analysis with R: a hands-on guide*. CRC Press, Boca Raton, first edition edition, 2022. ISBN 9781003107347.

Cochrane. Cochrane Handbook for Systematic Reviews of Interventions, 2023. URL https://training.cochrane.org/handbook/current.

Evangelos Evangelou and Areti Angeliki Veroniki, editors. *Meta-Research: Methods and Protocols*, volume 2345 of *Methods in Molecular Biology*. Springer US, New York, NY, 2022. ISBN 9781071615652 9781071615669. doi: 10.1007/978-1-0716-1566-9. URL https://link.springer.com/10.1007/978-1-0716-1566-9.

Timothy Hugh Barker, Celina Borges Migliavaca, Cinara Stein, Verônica Colpani, Maicon Falavigna, Edoardo Aromataris, and Zachary Munn. Conducting proportional meta-analysis in different types of systematic reviews: a guide for synthesisers of evidence. *BMC Medical Research Methodology*, 21(1):189, September 2021. ISSN 1471-2288. doi: 10.1186/s12874-021-01381-z. URL https://doi.org/10.1186/s12874-021-01381-z.

James Gareth, Daniel Witten, Trevor Hastie, and Robert Tibshirani, editors. *An Introduction to Statistical Learning*, volume 2345. Springer US, Los Angeles, LA, 2013. ISBN 9781461471370 9781461471387. doi: 10.1007/978-1-4614-7138-7. URL https://www.statlearning.com/.

Peter Mell, Karen Kent, and Joseph Nusbaum. *Index of /kashmiri/nist*. NIST, November 2005. URL https://profsite.um.ac.ir/kashmiri/nist/SP800-83.pdf.

Nirav Bhojani. Malware Analysis. *research Gate*, 2014. doi: 10.13140/2.1.4750.6889. URL http://rgdoi.net/10.13140/2.1.4750.6889.

Rabia Tahir. A Study on Malware and Malware Detection Techniques. *International Journal of Education and Management Engineering*, 8(2):20–30, March 2018. ISSN 23053623, 23058463. doi: 10.5815/ijeme.2018.02.03. URL http://www.mecs-press.org/ijeme/ijeme-v8-n2/v8n2-3.html.

Rami Sihwail, Khairuddin Omar, and Khairul Akram Zainol Ariffin. A Survey on Malware Analysis Techniques: Static, Dynamic, Hybrid and Memory Analysis. *International Journal on Advanced Science, Engineering and Information Technology*, 8(4-2):1662–1671, 2018. ISSN 2088-5334. URL http://ijaseit.insightsociety.org/index.php?option=com_content&view=article&id=9&Itemid=1&article_id=6827.

Sahil Sehrawat and Dr. Dinesh Singh. Malware and Malware Detection Techniques: A Survey. *International Journal for Research in Applied Science and Engineering Technology*, 10(5):3947–3953, May 2022. ISSN 23219653. doi: 10.22214/ijraset.2022.43287. URL https://www.ijraset.com/best-journal/malware-and-malware-detection-techniques-a-survey.

Sanjay K. Sahay, Ashu Sharma, and Hemant Rathore. Evolution of Malware and Its Detection Techniques. In Milan Tuba, Shyam Akashe, and Amit Joshi, editors, *Information and Communication Technology for Sustainable Development*, Advances in Intelligent Systems

and Computing, pages 139–150, Singapore, 2020. Springer. ISBN 9789811371660. doi: 10.1007/978-981-13-7166-0_14.

Eduardo De O. Andrade, José Viterbo, Joris Guérin, and Flavia Bernardini. Malware classification using word embeddings algorithms and long-short term memory networks. *Computational Intelligence*, 38(5):1802–1830, July 2022. ISSN 0824-7935, 1467-8640. doi: 10.1111/coin.12543. URL https://onlinelibrary.wiley.com/doi/10.1111/coin.12543.

Xiaolu Zhang, Frank Breitinger, Engelbert Luechinger, and Stephen O'Shaughnessy. Android application forensics: A survey of obfuscation, obfuscation detection and deobfuscation techniques and their impact on investigations. *Forensic Science International: Digital Investigation*, 39:301285, December 2021a. ISSN 2666-2817. doi: 10.1016/j.fsidi.2021.301285. URL https://www.sciencedirect.com/science/article/pii/S2666281721002031.

S. Sibi Chakkaravarthy, D. Sangeetha, and V. Vaidehi. A Survey on malware analysis and mitigation techniques. *Computer Science Review*, 32:1–23, May 2019. ISSN 1574-0137. doi: 10.1016/j.cosrev.2019.01.002. URL https://www.sciencedirect.com/science/article/pii/S1574013718301114.

Jinxin Liu, Michele Nogueira, Johan Fernandes, and Burak Kantarci. Adversarial Machine Learning: A Multilayer Review of the State-of-the-Art and Challenges for Wireless and Mobile Systems. *IEEE Communications Surveys & Tutorials*, 24(1):123–159, 2022. ISSN 1553-877X. doi: 10.1109/COMST.2021.3136132.

Kamran Shaukat, Suhuai Luo, and Vijay Varadharajan. A novel method for improving the robustness of deep learning-based malware detectors against adversarial attacks. *Engineering Applications of Artificial Intelligence*, 116:105461, November 2022. ISSN 0952-1976. doi: 10.1016/j.engappai.2022.105461. URL https://www.sciencedirect.com/science/article/pii/S0952197622004511.

Balram Yadav and Sanjiv Tokekar. Deep Learning in Malware Identification and Classification. In Mark Stamp, Mamoun Alazab, and Andrii Shalaginov, editors, *Malware Analysis Using Artificial Intelligence and Deep Learning*, pages 163–205. Springer International Publishing, Cham, 2021. ISBN 9783030625825. doi: 10.1007/978-3-030-62582-5_6. URL https://doi.org/10.1007/978-3-030-62582-5_6.

Seth Weidman. *Deep learning from scratch: building with Python from first principles*. O'Reilly Media, Inc, Sebastopol, CA, first edition edition, 2019. ISBN 9781492041412. OCLC: on1091365521.

Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Inc, Beijing [China] ; Sebastopol, CA, second edition edition, 2019. ISBN 9781492032649.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Gerard Urrútia and Xavier Bonfill. Declaración PRISMA: una propuesta para mejorar la publicación de revisiones sistemáticas y metaanálisis. *Medicina Clínica*, 135(11):507–511, October 2010. ISSN 0025-7753. doi: 10.1016/j.medcli.2010.01.015. URL https://www.sciencedirect.com/science/article/pii/S0025775310001454.

Yanfang Ye, Lingwei Chen, Shifu Hou, William Hardy, and Xin Li. DeepAM: a heterogeneous deep learning framework for intelligent malware detection. *Knowledge and Information Systems*, 54(2):265–285, May 2017. ISSN 0219-3116. doi: 10.1007/s10115-017-1058-9. URL https://doi.org/10.1007/s10115-017-1058-9.

Jixin Zhang, Zheng Qin, Hui Yin, Lu Ou, and Kehuan Zhang. A feature-hybrid malware variants detection using CNN based opcode embedding and BPNN based API embedding. *Computers & Security*, 84:376–392, July 2019. ISSN 0167-4048. doi: 10.1016/j.cose.2019.04.005. URL https://www.sciencedirect.com/science/article/pii/S0167404818312902.

Xin Ma, Shize Guo, Wei Bai, Jun Chen, Shiming Xia, and Zhisong Pan. An API Semantics-Aware Malware Detection Method Based on Deep Learning. *Security and Communication Networks*, 2019:e1315047, November 2019. ISSN 1939-0114. doi: 10.1155/2019/1315047. URL https://www.hindawi.com/journals/scn/2019/1315047/.

Dainius Čeponis and Nikolaj Goranin. Investigation of Dual-Flow Deep Learning Models LSTM-FCN and GRU-FCN Efficiency against Single-Flow CNN Models for the Host-Based Intrusion and Malware Detection Task on Univariate Times Series Data. *Applied Sciences*, 10(7):2373, March 2020. ISSN 2076-3417. doi: 10.3390/app10072373. URL https://www.mdpi.com/2076-3417/10/7/2373.

Seungho Jeon and Jongsub Moon. Malware-Detection Method with a Convolutional Recurrent Neural Network Using Opcode Sequences. *Information Sciences*, 535:1–15, October 2020. ISSN 0020-0255. doi: 10.1016/j.ins.2020.05.026. URL https://www.sciencedirect.com/science/article/pii/S0020025520304217.

Cengiz Acarturk, Melih Sirlanci, Pinar Gurkan Balikcioglu, Deniz Demirci, Nazenin Sahin, and Ozge Acar Kucuk. Malicious Code Detection: Run Trace Output Analysis by LSTM. *IEEE Access*, 9:9625–9635, January 2021. ISSN 2169-3536. doi: 10.1109/ACCESS.2021.3049200.

Robertas Damaševičius, Algimantas Venčkauskas, Jevgenijus Toldinas, and Šarūnas Grigaliūnas. Ensemble-Based Classification Using Neural Networks and Machine Learning Models for Windows PE Malware Detection. *Electronics*, 10(4):485, February 2021. ISSN

2079-9292. doi: 10.3390/electronics10040485. URL https://www.mdpi.com/2079-9292/10/4/485.

Jian Jiang and Fen Zhang. Detecting Portable Executable Malware by Binary Code Using an Artificial Evolutionary Fuzzy LSTM Immune System. *Security and Communication Networks*, 2021:e3578695, July 2021. ISSN 1939-0114. doi: 10.1155/2021/3578695. URL https://www.hindawi.com/journals/scn/2021/3578695/.

Rong Wang, Cong Tian, and Lin Yan. Malware Detection Using CNN via Word Embedding in Cloud Computing Infrastructure. *Scientific Programming*, 2021:e8381550, September 2021. ISSN 1058-9244. doi: 10.1155/2021/8381550. URL https://www.hindawi.com/journals/sp/2021/8381550/.

Donghai Tian, Qianjin Ying, Xiaoqi Jia, Rui Ma, Changzhen Hu, and Wenmao Liu. MDCHD: A novel malware detection method in cloud using hardware trace and deep learning. *Computer Networks*, 198:108394, October 2021. ISSN 1389-1286. doi: 10.1016/j.comnet.2021.108394. URL https://www.sciencedirect.com/science/article/pii/S1389128621003728.

Xiaohui Chen, Zhiyu Hao, Lun Li, Lei Cui, Yiran Zhu, Zhenquan Ding, and Yongji Liu. CruParamer: Learning on Parameter-Augmented API Sequences for Malware Detection. *IEEE Transactions on Information Forensics and Security*, 17:788–803, February 2022. ISSN 1556-6021. doi: 10.1109/TIFS.2022.3152360.

Young Jae Kim, Chan-Hyeok Park, and MyungKeun Yoon. FILM: Filtering and Machine Learning for Malware Detection in Edge Computing. *Sensors*, 22(6):2150, March 2022. ISSN 1424-8220. doi: 10.3390/s22062150. URL https://www.mdpi.com/1424-8220/22/6/2150.

Ce Li, Qiujian Lv, Ning Li, Yan Wang, Degang Sun, and Yuanyuan Qiao. A novel deep framework for dynamic malware detection based on API sequence intrinsic features. *Computers & Security*, 116:102686, May 2022. ISSN 0167-4048. doi: 10.1016/j.cose.2022.102686. URL https://www.sciencedirect.com/science/article/pii/S0167404822000840.

Donghai Tian, Runze Zhao, Rui Ma, Xiaoqi Jia, Qi Shen, Changzhen Hu, and Wenmao Liu. MDCD: A malware detection approach in cloud using deep learning. *Transactions on Emerging Telecommunications Technologies*, 33(11), June 2022. ISSN 2161-3915, 2161-3915. doi: 10.1002/ett.4584. URL https://onlinelibrary.wiley.com/doi/10.1002/ett.4584.

Cho Do Xuan, D. T. Huong, and Duc Duong. New approach for APT malware detection on the workstation based on process profile. *Journal*

of Intelligent & Fuzzy Systems, 43(4):4815–4834, August 2022. ISSN 1064-1246. doi: 10.3233/JIFS-212880. URL https://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs212880.

Vinayakumar Ravi, Mamoun Alazab, Shymalagowri Selvaganapathy, and Rajasekhar Chaganti. A Multi-View attention-based deep learning framework for malware detection in smart healthcare systems. *Computer Communications*, 195:73–81, November 2022. ISSN 0140-3664. doi: 10.1016/j.comcom.2022.08.015. URL https://www.sciencedirect.com/science/article/pii/S0140366422003231.

Abdullah I. A. Alzahrani, Manel Ayadi, Mashael M. Asiri, Amal Al-Rasheed, and Amel Ksibi. Detecting the Presence of Malware and Identifying the Type of Cyber Attack Using Deep Learning and VGG-16 Techniques. *Electronics*, 11(22):3665, November 2022. ISSN 2079-9292. doi: 10.3390/electronics11223665. URL https://www.mdpi.com/2079-9292/11/22/3665.

C. Catalano, A. Chezzi, M. Angelelli, and F. Tommasi. Deceiving AI-based malware detection through polymorphic attacks. *Computers in Industry*, 143:103751, December 2022. ISSN 0166-3615. doi: 10.1016/j.compind.2022.103751. URL https://www.sciencedirect.com/science/article/pii/S0166361522001488.

Chunlai Du, Ying Tong, Xiaohui Chen, Yongji Liu, Zhenquan Ding, Huixuan Xu, Qingyun Ran, Yi Zhang, Lingxiang Meng, Lei Cui, and Zhiyu Hao. Toward Detecting Malware Based on Process-Aware Behaviors. *Security and Communication Networks*, 2023:e6447655, January 2023. ISSN 1939-0114. doi: 10.1155/2023/6447655. URL https://www.hindawi.com/journals/scn/2023/6447655/.

Zhenlong Yuan, Yongqiang Lu, and Yibo Xue. Droiddetector: android malware characterization and detection using deep learning. *Tsinghua Science and Technology*, 21(1):114–123, February 2016. ISSN 1007-0214. doi: 10.1109/TST.2016.7399288.

Dina Saif, S. M. El-Gokhy, and E. Sallam. Deep Belief Networks-based framework for malware detection in Android systems. *Alexandria Engineering Journal*, 57(4):4049–4057, December 2018. ISSN 1110-0168. doi: 10.1016/j.aej.2018.10.008. URL https://www.sciencedirect.com/science/article/pii/S1110016818301996.

TaeGuen Kim, BooJoong Kang, Mina Rho, Sakir Sezer, and Eul Gyu Im. A Multimodal Deep Learning Method for Android Malware Detection Using Various Features. *IEEE Transactions on Information Forensics and Security*, 14(3):773–788, March 2019. ISSN 1556-6021. doi: 10.1109/TIFS.2018.2866319.

Aziz Alotaibi. Identifying Malicious Software Using Deep Residual Long-Short Term Memory. *IEEE Access*, 7:163128–163137, November 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2951751.

Mohammed K. Alzaylaee, Suleiman Y. Yerima, and Sakir Sezer. DL-Droid: Deep learning based android malware detection using real devices. *Computers & Security*, 89:101663, February 2020. ISSN 0167-4048. doi: 10.1016/j.cose.2019.101663. URL https://www.sciencedirect.com/science/article/pii/S0167404819300161.

Zhongru Ren, Haomin Wu, Qian Ning, Iftikhar Hussain, and Bingcai Chen. End-to-end malware detection for android IoT devices using deep learning. *Ad Hoc Networks*, 101: 102098, April 2020. ISSN 1570-8705. doi: 10.1016/j.adhoc.2020.102098. URL https://www.sciencedirect.com/science/article/pii/S1570870519310984.

Xinjun Pei, Long Yu, and Shengwei Tian. AMalNet: A deep learning framework based on graph convolutional networks for malware detection. *Computers & Security*, 93: 101792, June 2020. ISSN 0167-4048. doi: 10.1016/j.cose.2020.101792. URL https://www.sciencedirect.com/science/article/pii/S0167404820300778.

Weina Niu, Rong Cao, Xiaosong Zhang, Kangyi Ding, Kaimeng Zhang, and Ting Li. OpCode-Level Function Call Graph Based Android Malware Classification Using Deep Learning. *Sensors*, 20(13):3645, June 2020. ISSN 1424-8220. doi: 10.3390/s20133645. URL https://www.mdpi.com/1424-8220/20/13/3645.

Nan Zhang, Yu-an Tan, Chen Yang, and Yuanzhang Li. Deep learning feature exploration for Android malware detection. *Applied Soft Computing*, 102:107069, April 2021b. ISSN 1568-4946. doi: 10.1016/j.asoc.2020.107069. URL https://www.sciencedirect.com/science/article/pii/S1568494620310073.

Stuart Millar, Niall McLaughlin, Jesus Martinez del Rincon, and Paul Miller. Multi-view deep learning for zero-day Android malware detection. *Journal of Information Security and Applications*, 58:102718, May 2021. ISSN 2214-2126. doi: 10.1016/j.jisa.2020.102718. URL https://www.sciencedirect.com/science/article/pii/S2214212620308577.

Recep Sinan Arslan. AndroAnalyzer: android malicious software detection based on deep learning. *PeerJ Computer Science*, 7:e533, May 2021. ISSN 2376-5992. doi: 10.7717/peerj-cs.533. URL https://peerj.com/articles/cs-533.

Wenhui Zhang, Nurbol Luktarhan, Chao Ding, and Bei Lu. Android Malware Detection Using TCN with Bytecode Image. *Symmetry*, 13(7):1107, June 2021c. ISSN 2073-8994. doi: 10.3390/sym13071107. URL https://www.mdpi.com/2073-8994/13/7/1107.

Zakeya Namrud, Sègla Kpodjedo, Chamseddine Talhi, Ahmed Bali, and Alvine Boaye Belle. Deep Learning Based Android Anomaly Detection Using a Combination of Vulnerabilities Dataset. *Applied Sciences*, 11(16):7538, August 2021. ISSN 2076-3417. doi: 10.3390/app11167538. URL https://www.mdpi.com/2076-3417/11/16/7538.

Duc V. Nguyen, Giang L. Nguyen, Thang T. Nguyen, Anh H. Ngo, and Giang T. Pham. MINAD: Multi-inputs Neural Network based on Application Structure for Android Malware Detection. *Peer-to-Peer Networking and Applications*, 15(1):163–177, September 2021. ISSN 1936-6450. doi: 10.1007/s12083-021-01244-w. URL https://doi.org/10.1007/s12083-021-01244-w.

Ke Kong, Zhichao Zhang, Zi-Yuan Yang, and Zhaoxin Zhang. FCSCNN: Feature centralized Siamese CNN-based android malware identification. *Computers & Security*, 112:102514, January 2022. ISSN 0167-4048. doi: 10.1016/j.cose.2021.102514. URL https://www.sciencedirect.com/science/article/pii/S0167404821003382.

Iman Almomani, Aala Alkhayer, and Walid El-Shafai. An Automated Vision-Based Deep Learning Model for Efficient Detection of Android Malware Attacks. *IEEE Access*, 10:2700–2720, January 2022. ISSN 2169-3536. doi: 10.1109/ACCESS.2022.3140341.

Zhiqiang Wang, Gefei Li, Zihan Zhuo, Xiaorui Ren, Yuheng Lin, and Jieming Gu. A Deep Learning Method for Android Application Classification Using Semantic Features. *Security and Communication Networks*, 2022:e1289175, February 2022. ISSN 1939-0114. doi: 10.1155/2022/1289175. URL https://www.hindawi.com/journals/scn/2022/1289175/.

Hasan Alkahtani and Theyazn H. H. Aldhyani. Artificial Intelligence Algorithms for Malware Detection in Android-Operated Mobile Devices. *Sensors*, 22(6):2268, March 2022. ISSN 1424-8220. doi: 10.3390/s22062268. URL https://www.mdpi.com/1424-8220/22/6/2268.

Pooja Yadav, Neeraj Menon, Vinayakumar Ravi, Sowmya Vishvanathan, and Tuan D. Pham. EfficientNet convolutional neural networks-based Android malware detection. *Computers & Security*, 115:102622, April 2022. ISSN 0167-4048. doi: 10.1016/j.cose.2022.102622. URL https://www.sciencedirect.com/science/article/pii/S0167404822000219.

Somayyeh Fallah and Amir Jalaly Bidgoly. Android malware detection using network traffic based on sequential deep learning models. *Software: Practice and Experience*, 52(9):1987–2004, June 2022. ISSN 0038-0644, 1097-024X. doi: 10.1002/spe.3112. URL https://onlinelibrary.wiley.com/doi/10.1002/spe.3112.

Rahul Yumlembam, Biju Issac, Seibu Mary Jacob, and Longzhi Yang. IoT-Based Android Malware Detection Using Graph Neural Network With Adversarial Defense. *IEEE Internet*

of Things Journal, 10(10):8432–8444, July 2022. ISSN 2327-4662. doi: 10.1109/JIOT.2022.3188583.

Yafei Wu, Jian Shi, Peicheng Wang, Dongrui Zeng, and Cong Sun. DeepCatra: Learning flow- and graph-based behaviours for Android malware detection. *IET Information Security*, 17(1):118–130, August 2022. ISSN 1751-8709, 1751-8717. doi: 10.1049/ise2.12082. URL https://onlinelibrary.wiley.com/doi/10.1049/ise2.12082.

Abdullah Talha Kabakus. DroidMalwareDetector: A novel Android malware detection framework based on convolutional neural network. *Expert Systems with Applications*, 206:117833, November 2022. ISSN 0957-4174. doi: 10.1016/j.eswa.2022.117833. URL https://www.sciencedirect.com/science/article/pii/S0957417422010922.

Vinayakumar Ravi and Rajasekhar Chaganti. EfficientNet deep learning meta-classifier approach for image-based android malware detection. *Multimedia Tools and Applications*, December 2022. ISSN 1573-7721. doi: 10.1007/s11042-022-14236-6. URL https://doi.org/10.1007/s11042-022-14236-6.

Esraa Saleh Alomari, Riyadh Rahef Nuiaa, Zaid Abdi Alkareem Alyasseri, Husam Jasim Mohammed, Nor Samsiah Sani, Mohd Isrul Esa, and Bashaer Abbuod Musawi. Malware Detection Using Deep Learning and Correlation-Based Feature Selection. *Symmetry*, 15 (1):123, January 2023. ISSN 2073-8994. doi: 10.3390/sym15010123. URL https://www.mdpi.com/2073-8994/15/1/123.

Omar A. Alzubi, Jafar A. Alzubi, Tareq Mahmod Alzubi, and Ashish Singh. Quantum Mayfly Optimization with Encoder-Decoder Driven LSTM Networks for Malware Detection and Classification Model. *Mobile Networks and Applications*, July 2023. ISSN 1572-8153. doi: 10.1007/s11036-023-02105-x. URL https://doi.org/10.1007/s11036-023-02105-x.

Ashwag Albakri, Fatimah Alhayan, Nazik Alturki, Saahirabanu Ahamed, and Shermin Shamsudheen. Metaheuristics with Deep Learning Model for Cybersecurity and Android Malware Detection and Classification. *Applied Sciences*, 13(4):2172, February 2023. ISSN 2076-3417. doi: 10.3390/app13042172. URL https://www.mdpi.com/2076-3417/13/4/2172.

James P. Hunter, Athanasios Saratzis, Alex J. Sutton, Rebecca H. Boucher, Robert D. Sayers, and Matthew J. Bown. In meta-analyses of proportion studies, funnel plots were found to be an inaccurate method of assessing publication bias. *Journal of Clinical Epidemiology*, 67(8):897–903, August 2014. ISSN 0895-4356. doi: 10.1016/j.jclinepi.2014.03.003. URL https://www.sciencedirect.com/science/article/pii/S0895435614000869.

Apéndice A

Código para la experimentación

En este apéndice vamos a mostrar el código desarrollado en R así como las posibles llamadas a dicho código.

A.1. Función desarrollada en R

```
# Instalacion de paquetes (en caso de no disponer de ellos
# descomentar la siguiente linea)
# install.packages(c("metafor", "meta"))

# Carga las librerias necesarias
library(metafor)
library(meta)

# Definicion de la funcion
funcion_metaanalisis <- function(plataforma, tipoAnalisis, medida,
metodo, ficheroDatos, idExperimento, influyentesAEliminar=c()) {

    ######################################################
    # EXTRACCION Y CARGA DE DATOS
    ######################################################

    # Extrae los datos del fichero
    datos = read.csv(ficheroDatos, header=T, sep=';')

    if (plataforma==-1){
        # Nos quedamos con todos los datos del fichero
        datosAnalisis = datos;
    } else if(plataforma==0){
        # Nos quedamos con los datos de la plataforma escogida
```

```
25          datosAnalisis = datos[datos["Plataforma"]=="Windows", ]
26      } else {
27          datosAnalisis = datos[datos["Plataforma"]=="Android", ]
28      }
29
30      if (tipoAnalisis!=-1){
31          # Nos quedamos con los datos de la plataforma
32          # y el tipo de analisis escogido
33          datosAnalisis = datosAnalisis[
34              datosAnalisis["TipoAnalisisNum"]==tipoAnalisis, ]
35      }
36
37      if(length(influyentesAEliminar)> 0) {
38          # Eliminamos los estudios influyentes en caso de existir
39          datosAnalisis = datosAnalisis[-influyentesAEliminar, ]
40      }
41
42      print(datosAnalisis)
43
44      if(medida == 0) {
45          xiMedida = t(datosAnalisis["TP"])[, ];
46          niMedida = t(datosAnalisis["CasosPositivos"])[, ];
47          mensajeMedida = "Sensibilidad";
48          mensajeCasos = "TP";
49          mensajeTotales = "Casos positivos (total)";
50      } else if (medida == 1){
51          xiMedida = t(datosAnalisis["TN"])[, ];
52          niMedida = t(datosAnalisis["CasosNegativos"])[, ];
53          mensajeMedida = "Especificidad";
54          mensajeCasos = "TN";
55          mensajeTotales = "Casos negativos (total)"
56      }
57
58      if(metodo==0 || metodo==3) {
59          transformacion = "PLO";
60          transformacionInversa = transf.ilogit;
61      } else if (metodo==1) {
62          transformacion = "PFT";
63          transformacionInversa = transf.ipft.hm;
64      } else if (metodo==2) {
65          transformacion = "PAS";
66          transformacionInversa = transf.iarcsin;
```

```
67      }
68
69      ######################################################
70      # ANALISIS
71      ######################################################
72
73      # ----------------------------------------------------
74      # PASO 1. Realizamos el calculo de la medida de efecto
75      # global o combinada.
76      #
77      # Vamos a usar la funcion "escalc" para calcular las
78      # medidas de efecto individuales (yi) y sus varianzas
79      # muestrales (vi).
80      # El calculo de las yi y las vi es como sigue
81      # (en el caso de logit, transformacion = ln, por ejemplo):
82      #       yi = transformacion ((xi/ni)/(1-xi/ni))
83      #   vi = 1/xi + 1/(ni-xi)
84      # Teniendo en cuenta que si la proporcion (xi/ni) = 1,
85      # sumara 0.5 a xi y a ni-xi, por lo que la proporcion usada
86      # seria p = (xi+0,5)/(ni+1)
87      #
88      # Posteriormente aplicamos la funcion "rma" (random effects
89      # metaanalysis) que crea la medida de efecto combinada con la
90      # transformacion aplicada (logit, doble-arcoseno, ...).
91      #
92      # Finalmente usamos "predict" para obtener la medida de efecto
93      # combinada para deshacer la transformacion aplicada y que el
94      # resultado sea una proporcion
95      # ----------------------------------------------------
96
97      if(metodo == 0 || metodo == 2) {
98          # mei = medidas de efecto individuales
99          mei = escalc(xi=xiMedida, ni=niMedida,
100                       measure=transformacion)
101         # mec = medida de efecto combinada transformada
102         mect = rma(yi, vi, data=mei, method="DL", level=95)
103         # mecp = medida de efecto combinada de proporciones
104         mecp = predict(mect, transf=transformacionInversa)
105     } else if (metodo == 1) {
106         # mei = medidas de efecto individuales
107         mei = escalc(xi=xiMedida, ni=niMedida,
108                      measure=transformacion, add=0)
```

```
109         # mec = medida de efecto combinada transformada
110         mect = rma(yi, vi, data=mei, method="DL", level=95)
111         # mecp = medida de efecto combinada de proporciones
112         mecp = predict(mect, transf=transformacionInversa,
113                        targ=list(ni=niMedida))
114     } else if (metodo == 3){
115         mect = rma.glmm(measure="PLO", xi=xiMedida, ni=niMedida,
116                         data=datosAnalisis)
117         mecp = predict(mect, transf=transf.ilogit)
118     }
119     print(mecp, digit=6)
120 
121     # ------------------------------------------------------
122     # PASO 2. Identificamos y cuantificamos la presencia de
123     # heterogeneidad
124     #
125     # Los valores mas significativos en los que nos tenemos
126     # que fijar son:
127     # tau^2 que es un estimador de la cantidad de
128     # heterogeneidad (tambien se nos da su IC al 95%)
129     #   Q estadistico que evalua la presencia de heterogeneidad
130     #   I^2 que cuantifica la cantidad de heterogeneidad
131     #   (Se asume que 25, 50, 75% indican baja, media y gran
132     #   heterogeneidad respectivamente)
133     # ------------------------------------------------------
134 
135     print(mect, digits=4)
136     if (metodo != 3) {
137         print(confint(mect, digits=2))
138     }
139 
140     # ------------------------------------------------------
141     # PASO 3. Obtenemos el forest plot
142     # ------------------------------------------------------
143 
144     mecp.summary = metaprop(xiMedida, niMedida, AutorAnyo,
145                             data=datosAnalisis, sm=transformacion)
146     if(metodo==3) {
147         precision = 0
148     } else {
149         precision = sqrt(mei$vi)
150     }
```

```
151
152     if (nrow(datosAnalisis)>30) {
153         png(file=paste('imagenes/Experimento_', idExperimento,
154                        '_forest_plot.png', sep=""),
155             width=2048,height=1080)
156     } else {
157         png(file=paste('imagenes/Experimento_', idExperimento,
158                        '_forest_plot.png', sep=""),
159             width=1024, height=720)
160     }
161
162     if(metodo==3) {
163         forest(mecp.summary, xlim=c(0.82, 1.1), rightcols=FALSE,
164                leftcols=c("studlab", "event", "n", "effect", "ci"),
165                leftlabs=c("Estudio", mensajeCasos, mensajeTotales,
166                           "Proporcion", "95% C.I."),
167                xlab=mensajeMedida, smlab="",
168                weight.study="random",
169                squaresize=0.5, col.square="navy",
170                col.square.lines="navy", col.diamond="maroon",
171                col.diamond.lines="maroon",
172                pooled.totals=FALSE, comb.fixed=FALSE,
173                fs.heatstat=10,
174                print.tau2=TRUE, print.Q=TRUE,
175                print.pval.Q=TRUE, print.I2=TRUE, digits=2)
176     } else {
177         forest(mecp.summary, xlim=c(0.82, 1.1), rightcols=FALSE,
178                leftcols=c("studlab", "event", "n", "effect", "ci"),
179                leftlabs=c("Estudio", mensajeCasos, mensajeTotales,
180                           "Proporcion", "95% C.I."),
181                xlab=mensajeMedida, smlab="",
182                weight.study="random",
183                squaresize=0.5, col.square="navy",
184                col.square.lines="navy", col.diamond="maroon",
185                col.diamond.lines="maroon",
186                pooled.totals=FALSE, comb.fixed=FALSE,
187                fs.heatstat=10,
188                print.tau2=TRUE, print.Q=TRUE,
189                print.pval.Q=TRUE, print.I2=TRUE,
190                digits=2, sortvar=precision)
191
192     dev.off();
```

```
193
194
195     # ------------------------------------------------------
196     # PASO 4. Obtenemos el funnel plot
197     # ------------------------------------------------------
198
199     png(file=paste('imagenes/Experimento_', idExperimento,
200                    '_funnel_plot.png', sep=""),
201         width=1024, height=720)
202
203     # funnel plot con proporcion logit en lugar de logodds y
204     # error estandar en eje y
205     if (metodo == 1) {
206         funnel(mect, yaxis="sei", atransf=transformacionInversa,
207                targ=list(ni=niMedida))
208     } else {
209         funnel(mect, yaxis="sei", atransf=transformacionInversa)
210     }
211
212     dev.off();
213
214     # ------------------------------------------------------
215     # PASO 5. Analizamos outliers (Solo en casos No GLMMM)
216     #
217     # En primer lugar vamos a usar leave-one-out, para revisar el
218     # impacto que tienen los estudios dejando uno fuera cada vez y
219     # recalculando valores
220     # Despues realizaremos un forest plot del resultado anterior.
221     # Cuanto mas alejado este el cuadrado de la linea de referencia
222     # mas pronunciado es el impacto
223     # de su falta en la suma de proporciones original
224     # Posteriormente comprobamos que estudios nos marca como
225     # influyentes (*)
226     # ------------------------------------------------------
227
228     if(metodo != 3) {
229         l1o=leave1out(mect)
230         l1o
231
232         yi = l1o$estimate; vi=l1o$se^2
233         png(file=paste('imagenes/Experimento_', idExperimento,
234                        '_forest_plot_one_out.png', sep=""),
```

```
            width=1024, height=720)

        if (metodo == 1) {
            forest(yi, vi, transf=transformacionInversa,
                   targ=list(ni=niMedida),
                    slab=paste(datosAnalisis$Autor,
                               datosAnalisis$Anyo, sep=","),
                    refline=mecp$pred,
                    xlab="Resumen de proporciones dejando fuera
                          cada estudio")
        } else {
            forest(yi, vi, transf=transformacionInversa,
                   slab=paste(datosAnalisis$Autor,
                              datosAnalisis$Anyo, sep=","),
                   refline=mecp$pred,
                   xlab="Resumen de proporciones dejando fuera
                         cada estudio")
        }
        dev.off();

        inf=influence(mect)
        print(inf);
        png(file=paste('imagenes/Experimento_', idExperimento,
                       '_influyentes.png', sep=""),
            width=1024, height=720)
        plot(inf);
        dev.off();
    }
}
```

A.2. Llamadas a la función desarrollada

A.2.1. Análisis global

Experimento 1

Metaanálisis de la sensibilidad de todos los artículos usando el método del inverso de la varianza con la transformación logit

```
funcion_metaanalisis(-1, -1, 0, 0, "Datos.csv", "000", )
funcion_metaanalisis(-1, -1, 0, 0, "Datos.csv", "000b", c(14))
funcion_metaanalisis(-1, -1, 0, 0, "Datos.csv", "000c", c(14, 33))
```

Experimento 2

Metaanálisis de la sensibilidad de todos los artículos usando el método del inverso de la varianza con la transformación FT

```
funcion_metaanalisis(-1, -1, 0, 1, "Datos.csv", "001", )
funcion_metaanalisis(-1, -1, 0, 1, "Datos.csv", "001b",
                     c(5, 6, 19, 31, 33, 36, 37, 38, 39, 43, 44))
```

Experimento 3

Metaanálisis de la sensibilidad de todos los artículos usando el método del inverso de la varianza con la transformación arcoseno

```
funcion_metaanalisis(-1, -1, 0, 2, "Datos.csv", "002", )
funcion_metaanalisis(-1, -1, 0, 1, "Datos.csv", "002b",
                     c(5, 6, 19, 31, 36, 37, 38, 39, 43, 44))
```

Experimento 4

Metaanálisis de la sensibilidad de todos los artículos usando el método GLMM con la transformación logit

```
funcion_metaanalisis(-1, -1, 0, 3, "Datos.csv", "003", )
```

Experimento 5

Metaanálisis de la especificidad de todos los artículos usando el método del inverso de la varianza con la transformación FT

```
funcion_metaanalisis(-1, -1, 1, 1, "Datos.csv", "011", )
```

A.2.2. Estratificación por plataforma

Experimento 6

Metaanálisis de la sensibilidad de los artículos de Windows usando el método del inverso de la varianza con la transformación FT

```
funcion_metaanalisis(0, -1, 0, 1, "Datos.csv", "101", )
funcion_metaanalisis(0, -1, 0, 1, "Datos.csv", "101b", c(19))
funcion_metaanalisis(0, -1, 0, 1, "Datos.csv", "101c",
                     c(6, 10, 11, 12, 13, 14, 15, 17, 19, 20))
funcion_metaanalisis(0, -1, 0, 1, "Datos.csv", "101d",
    c(2, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18, 19, 20))
```

Experimento 7

Metaanálisis de la sensibilidad de los artículos de Android usando el método del inverso de la varianza con la transformación FT

```
funcion_metaanalisis(1, -1, 0, 1, "Datos.csv", "201", )
```

Experimento 8

Metaanálisis de la especificidad de los artículos de Windows usando el método del inverso de la varianza con la transformación FT

```
funcion_metaanalisis(0, -1, 1, 1, "Datos.csv", "111", )
funcion_metaanalisis(0, -1, 1, 1, "Datos.csv", "111b", c(7, 15))
funcion_metaanalisis(0, -1, 1, 1, "Datos.csv", "111c", c(7, 15, 19))
```

Experimento 9

Metaanálisis de la especificidad de los artículos de Android usando el método del inverso de la varianza con la transformación FT

```
funcion_metaanalisis(1, -1, 1, 1, "Datos.csv", "211", )
funcion_metaanalisis(1, -1, 1, 1, "Datos.csv", "211b", c(25))
funcion_metaanalisis(1, -1, 1, 1, "Datos.csv", "211c", c(16, 25))
```

A.2.3. Estratificación por plataforma y tipo de análisis

Experimento 10

Metaanálisis de la sensibilidad de los artículos de Windows con tipo de análisis estático usando el método del inverso de la varianza con la transformación FT

```
funcion_metaanalisis(0, 0, 0, 1, "Datos.csv", "1101", )
funcion_metaanalisis(0, 0, 0, 1, "Datos.csv", "1101b", c(10))
funcion_metaanalisis(0, 0, 0, 1, "Datos.csv", "1101c", c(8, 10))
```

Experimento 11

Metaanálisis de la sensibilidad de los artículos de Windows con tipo de análisis dinámico usando el método del inverso de la varianza con la transformación FT

```
funcion_metaanalisis(0, 1, 0, 1, "Datos.csv", "2101", )
funcion_metaanalisis(0, 1, 0, 1, "Datos.csv", "2101b", c(1))
funcion_metaanalisis(0, 1, 0, 1, "Datos.csv", "2101c", c(1, 2, 3))
funcion_metaanalisis(0, 1, 0, 1, "Datos.csv", "2101d", c(1, 2, 3, 6))
```

Experimento 12

Metaanálisis de la sensibilidad de los artículos de Windows con tipo de análisis híbrido usando el método del inverso de la varianza con la transformación FT

```
funcion_metaanalisis(0, 2, 0, 1, "Datos.csv", "3101", )
funcion_metaanalisis(0, 2, 0, 1, "Datos.csv", "3101b", c(3))
```

Experimento 13

Metaanálisis de la sensibilidad de los artículos de Android con tipo de análisis estático usando el método del inverso de la varianza con la transformación FT

```
funcion_metaanalisis(1, 0, 0, 1, "Datos.csv", "1201", )
```

Experimento 14

Metaanálisis de la sensibilidad de los artículos de Android con tipo de análisis híbrido usando el método del inverso de la varianza con la transformación FT

```
funcion_metaanalisis(1, 2, 0, 1, "Datos.csv", "3201", )
 funcion_metaanalisis(1, 2, 0, 1, "Datos.csv", "3201b", c(6))
funcion_metaanalisis(1, 2, 0, 1, "Datos.csv", "3201c", c(6, 7))
funcion_metaanalisis(1, 2, 0, 1, "Datos.csv", "3201d", c(1, 6, 7))
```

Experimento 15

Metaanálisis de la especificidad de los artículos de Windows con tipo de análisis estático usando el método del inverso de la varianza con la transformación FT

```
funcion_metaanalisis(0, 0, 1, 1, "Datos.csv", "1111", )
funcion_metaanalisis(0, 0, 1, 1, "Datos.csv", "1111b", c(6, 8))
```

Experimento 16

Metaanálisis de la especificidad de los artículos de Windows con tipo de análisis dinámico usando el método del inverso de la varianza con la transformación FT

```
funcion_metaanalisis(0, 1, 1, 1, "Datos.csv", "2111", )
funcion_metaanalisis(0, 1, 1, 1, "Datos.csv", "2111b", c(6))
funcion_metaanalisis(0, 1, 1, 1, "Datos.csv", "2111c", c(4, 6))
```

Experimento 17

Metaanálisis de la especificidad de los artículos de Windows con tipo de análisis híbrido usando el método del inverso de la varianza con la transformación FT

```
funcion_metaanalisis(0, 2, 1, 1, "Datos.csv", "3111", )
```

Experimento 18

Metaanálisis de la especificidad de los artículos de Android con tipo de análisis estático usando el método del inverso de la varianza con la transformación FT

```
funcion_metaanalisis(1, 0, 1, 1, "Datos.csv", "1211", )
funcion_metaanalisis(1, 0, 1, 1, "Datos.csv", "1211b", c(17))
funcion_metaanalisis(1, 0, 1, 1, "Datos.csv", "1211c", c(13, 17))
```

Experimento 19

Metaanálisis de la especificidad de los artículos de Android con tipo de análisis híbrido usando el método del inverso de la varianza con la transformación FT

```
funcion_metaanalisis(1, 2, 1, 1, "Datos.csv", "3211", )
funcion_metaanalisis(1, 2, 1, 1, "Datos.csv", "3211b", c(1))
funcion_metaanalisis(1, 2, 1, 1, "Datos.csv", "3211c", c(1, 5))
```