UNIVERSIDAD DE MURCIA

ESCUELA INTERNACIONAL DE DOCTORADO

Diseño de un Modelo Colaborativo de Intercambio de Anuncios entre Redes de Publicidad para Optimizar la Rentabilidad

D. Luis Miralles Pechuán

2017

UNIVERSIDAD DE MURCIA

D. Fernando Jiménez Barrionuevo, Profesor titular del Departamento de Ingeniería de la Información y las Comunicaciones de la Universidad de Murcia (España) y

D. José Manuel García Carrasco, Catedrático de Universidad del Departamento de Ingeniería y Tecnología de Computadores de la Universidad de Murcia

AUTORIZAN:

La presentación de la tesis doctoral titulada *«Diseño de un modelo colaborativo de intercambio de anuncios entre redes de publicidad para optimizar la rentabilidad»*, realizada por D. Luis Miralles Pechuán, bajo su inmediata dirección y supervisión, y que presenta para la obtención del grado de Doctor por la Universidad de Murcia.

En Murcia, a 9 de Febrero de 2017.

Fdo.: Dr. Fernando Jiménez Barrionuevo

Fdo.: Dr. José Manuel García Carrasco

A mis padres, a la Universidad Panamericana
y a mis directores de tesis.

Esta tesis ha sido escrita usando LaTeX.

Agradecimientos

Quiero dar las gracias al Dr. Fernando Jiménez Barrionuevo por toda la ayuda que me ha prestado, por su gran dedicación, por su compromiso y por su gran profesionalidad. También me siento en deuda con el Dr. José Manuel García Carrasco por todo su apoyo, por sus acertados consejos y por todas las facilidades que me ha dado a lo largo de la tesis doctoral.

Debo agradecer a la Universidad Panamericana de México, especialmente al Dr. Santiago García Álvarez y al Dr. Félix Martínez Ríos el haberme dado una oportunidad de trabajar en la Universidad Panamericana. Así como el haber estado pendientes de mí a lo largo de estos años. Esto me ha permitido terminar la tesis doctoral. Sin su ayuda, desarrollar esta tesis doctoral no habría sido posible.

Estoy muy agradecido al Dr. Alejandro Ordoñez, actual director de la facultad de ingeniería, por su espíritu emprendedor y su apertura a nuevos puntos de vista. No puedo olvidarme de la Dra. Lourdes Martínez Villaseñor por su buen humor, por su paciencia y por haberme impulsado a nuevos retos. Tampoco puedo olvidar de la Dra. Victoria Carreras Cruz por su disponibilidad, por su trato amable y por su capacidad resolutiva para abordar cualquier situación.

Aprovecho para mencionar a aquellas personas que han colaborado conmigo en los artículos que he publicado: Al Dr. Hiram Ponce Espinosa por su enorme talento y por su excelente calidad como profesional y como persona. A la Dra. Dafne Rosso Pelayo por su rigor científico, por su cordialidad y por su laboriosidad.

También agradecer el apoyo económico para este trabajo que ha sido apoyado conjuntamente por la Fundación Séneca (Agencia Regional de Ciencia y Tecnología, Región de Murcia) en virtud de concesión de fondos 15290/PI/2010, y el MINECO español y MEC español, así como la Comisión Europea FEDER bajo los proyectos TIN2012-31345 y TIN2015-66972-C5-3-R.

Finalmente, agradezco a mis padres su buen ejemplo, los valores que me han sabido transmitir y su apoyo incondicional. También a todos mis hermanos por toda la ayuda que me han prestado.

Resumen

En los inicios de Internet se crearon muchas redes publicitarias, pero a pesar de que la demanda de campañas *online* ha aumentado, el número de redes publicitarias ha ido disminuyendo. Las grandes multinacionales como Google o Facebook han ido paulatinamente acaparando el mercado. Sin embargo, las pequeñas redes publicitarias están en una situación delicada, principalmente porque carecen de sistemas eficientes para la detección de fraude *online* y porque no tienen un número de visitas suficiente para ofrecer a los anunciantes campañas con *microtargeting*. Es decir, campañas dirigidas a un pequeño grupo de usuarios con intereses comunes.

Esta tesis doctoral está enfocada a desarrollar un modelo de intercambio de anuncios para potenciar la creación y la mejora del rendimiento de las pequeñas redes. De forma que no solamente se evite que muchas pequeñas redes sigan desapareciendo sino que también se impulse el crecimiento y la creación de nuevas plataformas publicitarias.

El modelo de intercambio de anuncios propuesto está basado en un nuevo enfoque que tiene en cuenta múltiples criterios para garantizar el bienestar de todos los roles implicados en el sistema publicitario. Junto al tradicional objetivo de obtener los máximos ingresos económicos en las campañas de publicidad, el modelo propuesto añade otros criterios como son garantizar la satisfacción de editores, anunciantes y redes publicitarias, así como evitar cualquier tipo de fraude en el ecosistema publicitario.

Para la selección de un anuncio se define una función objetivo donde se representa cada uno de los criterios establecidos y donde cada criterio está ponderado mediante un coeficiente. Se ha definido una métrica para medir el rendimiento del modelo que consiste en restar a los ingresos del modelo un conjunto de penalizaciones cuando un criterio no se cumple. Los coeficientes asociados a cada objetivo se optimizan mediante un algoritmo genético para maximizar la métrica definida. También se compara esta metodología con el conocido método de selección de anuncios *Generalized Second Price* (GSP).

Para poder seleccionar un anuncio, lo más importante es determinar el valor de económico de dicho anuncio. Esta tesis ha desarrollado una metodología para determinar el valor de un anuncio en las redes CPM, CPC y CPA, que son las formas de pago más extendidas hoy

en día. Dicha metodología se basa en un sistema compuesto por cuatro módulos. El Módulo 1 calcula la probabilidad de que el anuncio sea *spam*. El Módulo 2 calcula el CTR, que es la probabilidad de que un usuario acceda a la página web de un editor y genere un clic en un anuncio. El Módulo 3 calcula la probabilidad de que se genere una venta a partir de una visita. Y el Módulo 4 calcula el valor de un anuncio según el modelo de pago escogido por el anunciante y el precio que está dispuesto a pagar.

Finalmente, se han optimizado algunos módulos mediante la selección de variables. También se han comparado algunos de los métodos de selección de variables más extendidos (RFE, NSGA-II, PCA y *Gain Ratio*) con ENORA. Estos algoritmos se utilizan en la predicción de ventas (regresión), así como para la predicción del CTR de un anuncio (clasificación). El método basado en el algoritmo ENORA no ha sido propuesto en esta tesis, pero ha sido utilizado, evaluado y testeado en profundidad en el contexto de predicción de ventas y en la estimación del CTR, sirviendo éstos como un excelente banco de pruebas para su validación como método eficaz para la selección de atributos.

Abstract

At the beginning of the Internet, many advertising networks were created. But despite the demand of online campaigns has increased, the number of advertising networks has been diminishing. The large multinationals such as Google or Facebook have been monopolizing gradually the market. Nevertheless, small advertising networks are in a delicate situation, mainly because they lack of efficient systems for detecting online fraud and they do not have sufficient number of visits to offer advertisers microtargeted campaigns. These campaigns are directed to a small user group with common interests.

This doctoral thesis is focused on developing an advertisement exchange model (AdX) to promote the creation of new advertising networks, and to improve the yield of the small ones. In such a way that, not only it is prevented that many small networks disappear but also the growth and the creation of new ones is encouraged.

The proposed AdX is based on a new approach that takes into account multiple criteria to guarantee the wellbeing of all involved roles in the advertising ecosystem. Besides the principal criterion of maximizing income, the proposed model includes other criteria such as guaranteeing publishers, advertisers and advertising networks satisfaction, as well as, avoiding fraud of any type.

For the selection of an advertisement, an objective function is defined where each criterion is represented and computed by means of a coefficient. A metric for evaluating the model performance has also been defined. The metric consists of subtracting from the model income a penalty for each criterion that is not fulfilled. The coefficient of each goal is optimized using a genetic algorithm in order to maximize the defined metric. The proposed methodology is compared with the well-known selection method Generalized Second Price (GSP).

In order to select an advert, it is required to calculate the advert value in economic terms. In this thesis, a methodology has been developed to determine the advert value in CPM, CPC and CPA payment methods, which are the most extended ones nowadays . The proposed methodology is based on a system composed of four modules. Module 1 calculates the probability that the advert is spam. Module 2 estimates the advert CTR, which it is the

probability that a user generates a click on an advert. Module 3 calculates the probability that a sale is generated from a user visit. Module 4 calculates the value of an advert according to the payment model chosen by the advertiser and the price he/she is willing to pay.

Finally, we have optimized some modules using feature selection methods. The performance of many important feature selection methods (RFE, NSGA-II, PCA and Gain Ratio) has been compared with the performance of ENORA. These algorithms are used in the prediction of sales (regression), as well as in estimating an advert CTR (classification). The method based on the ENORA algorithm has not been proposed in this thesis, but it has been used, evaluated and tested in depth in the context of sales forecast as in the CTR estimation. All these comparisons have served as an excellent test be d for validation of ENORA as an effective method of attributes selection.

Índice general

Índice de figuras

Índice de tablas

Índice de algoritmos

Parte I

Motivación y fundamentos

Capítulo 1

Motivación

1.1. Introducción

Internet es uno de los inventos más revolucionarios de la historia de la humanidad. Ha provocado cambios radicales y profundos en la conducta de vida de las personas a una velocidad nunca antes vista. Esto ha supuesto un ecosistema ideal para que muchas empresas desarrollen su negocio. El número de empresas que ha florecido y su facturación aumenta paulatinamente año tras año. Dentro de este nuevo mundo, la publicidad juega un papel fundamental pues permite a las empresas darse a conocer y captar nuevos clientes.

La publicidad en Internet ha evolucionado en los últimos años dando lugar a un gran mercado global [1]. Este gran mercado puede considerarse como una subasta global donde los anunciantes compiten entre sí por mostrar sus anuncios. Esto se conoce comúnmente como apuestas en tiempo real (*Real-time bidding*) (RTB)[1].

Sobre este tema se han investigado varias líneas. Por ejemplo, Castelluccia Claude [4] intenta encontrar un equilibrio entre la información que las redes conocen sobre los usuarios y el respeto a su privacidad. Cuanta más información tengan las redes acerca de los usuarios, más fácil les resultará encontrar el mejor anuncio para cada usuario.

Otras investigaciones se centran en integrar en un mismo Modelo de Intercambio de Anuncios (MIA) el modelo de pago CPC en el lado de los anunciantes y los modelos CPM en el lado del editor. En otras palabras, algunos editores quieren cobrar independientemente de si los usuarios hacen clic o no en el anuncio. Y algunos anunciantes quieren pagar únicamente si se genera un clic. Este problema puede resolverse utilizando un rol intermedio llamado *"arbitrage"* [5]. Este rol paga por adelantado haciendo una predicción del número de clics y su éxito depende de qué tan precisas sean dichas predicciones.

[1] Los anunciantes hacen una oferta para mostrar sus anuncios en el sitio web de los editores. Si un anunciante gana la puja, su anuncio se muestra instantáneamente en el sitio web del editor [2, 3].

Esta investigación comparte el enfoque de Balseiro [6], dado que considera más importante desarrollar un sistema orientado a la satisfacción de todos los roles involucrados, que desarrollar un sistema orientado a seleccionar el anuncio más rentable. Centrarse demasiado en el aspecto económico puede causar que los anunciantes hagan campañas desastrosas en términos de rendimiento económico y por lo tanto, que se pierdan clientes. También puede provocar que se disparen las impresiones de anuncios *spam*[2] o que haya anunciantes sin apenas impresiones.

El objetivo de esta tesis doctoral consiste en proponer un MIA que cumpla con los objetivos necesarios para que el ecosistema de publicidad funcione correctamente. Para ello, se representa a cada objetivo en la fórmula de selección de un anuncio y posteriormente se asigna un peso a cada objetivo. Los mejores pesos son los que maximizan los ingresos y a la vez reducen al mínimo la suma de todas las penalizaciones. Las penalizaciones son sanciones económicas que se aplican cuando no se logra un objetivo. Cuanto menos se cumpla un objetivo mayor será la penalización asociada a dicho objetivo. También se propone una metodología para encontrar los pesos óptimos en la fórmula de selección de anuncio.

La mejor configuración de pesos no se puede calcular en tiempo real porque el tiempo requerido para el cálculo de los pesos es mucho mayor que el tiempo requerido para mostrar un anuncio. Sin embargo, el valor de estos pesos puede calcularse cada cierto tiempo y los pesos pueden ser actualizados. Esta metodología es capaz de encontrar los pesos óptimos utilizando un algoritmo genético.

1.1.1. Creación de Internet

Si se tuviera que enumerar los inventos que más han influido en la forma de vida de las personas, no cabe duda de que Internet sería uno de ellos [8]. Internet ha supuesto un antes y un después en muchas de las principales actividades que se realizan en la vida cotidiana. La Red proporciona a la sociedad un acceso instantáneo a la información y además, permite la comunicación entre personas de todo el mundo [9].

Existen personas que viven gracias a las visitas en sus vídeos de Youtube, personas que mediante sus blogs influyen en la opinión de toda una sociedad y personas que siguen a los famosos a través de Twitter.

Cada vez hay un mayor número de usuarios en la Red y estos pasan cada día más tiempo conectados. El crecimiento de la oferta de productos y servicios en este medio es continuo.

[2]Los anuncios *spam* desvían a usuarios hacia sitios infectados con virus. También hacen falsas ofertas de productos con un precio por debajo del valor de mercado a fin de engañar a usuarios para obtener datos confidenciales tales códigos de la cuenta bancaria, contraseña del correo electrónico o información personal. En otros casos, los anunciantes de tipo *spam* se orientan a instalar un *malware* o un programa troyano en los ordenadores de los usuarios[7].

Comprar billetes de avión, hacer de *broker* en bolsa, comprar un televisor, ver películas *online*, ver la televisión, escuchar la radio, leer noticias, escribir correos electrónicos, chatear, hablar por videoconferencia son solo algunas de las actividades que permite la Red. Los dispositivos electrónicos tales como tabletas, móviles o portátiles han acompañado este cambio haciendo de Internet algo accesible, rápido y económico.

Internet ha supuesto un enorme impacto en el desarrollo científico y tecnológico dado que los equipos pueden estar formados por personas de todo el mundo y el conocimiento se puede compartir con el resto de la comunidad científica de manera casi inmediata. Por otra parte, el nacimiento de los motores de búsqueda permite a cualquier persona encontrar información mucho más fácilmente de lo que hasta ahora era posible, y desde la comodidad de su propio escritorio.

Cada año más personas acceden a la Red, y éstas a su vez, pasan más horas conectadas. Los dispositivos móviles están continuamente intercambiando información con los servidores, lo cual, permite a muchas personas estar permanentemente actualizados.

La Arpanet fue la antecesora de lo que hoy en día se conoce como Internet. Su nacimiento data de finales de los años 60, aunque cuando realmente se dio a conocer mundialmente fue en 1983. Tan solo dos años más tarde muchos ciudadanos de Estados Unidos se conectaron a esta red.

En los comienzos de Internet se requería que los usuarios tuvieran conocimientos muy avanzados para poder navegar. Además, las conexiones y las transferencias de archivos eran sumamente lentas y estaban muy limitadas. Por esas fechas, el ingeniero británico Tim Berners-Lee, uno de los padres de Internet, creó el *World Wide Web* (WWW). El WWW es un sistema de páginas web interconectadas y accesibles mediante hipertexto para facilitar el acceso a los usuarios.

A través de Internet se pueden realizar múltiples actividades sin restricciones geográficas o de horario. Entre las actividades más habituales en la Red están: escuchar música, intercambiar correos electrónicos, leer noticias, buscar información sobre un tema o tener conversaciones por vídeo conferencia. Algunas de estas actividades eran impensables hasta hace pocas décadas [10].

Los buscadores y los navegadores han sido dos elementos vitales para facilitar a los ciudadanos el acceso a Internet. La enorme cantidad de información (enciclopedias, artículos científicos, direcciones, vídeos, etc.) y la gran diversidad de servicios a ningún coste (email, foros, chats, etc.), han hecho que la Red se convierta en algo imprescindible en la vida de la mayoría de los ciudadanos.

Las aplicaciones web como Facebook y Twitter permiten saber qué están haciendo nuestros amigos. Mediante aplicaciones como Skype o WhatsApp existe un contacto permanente

con las personas de un entorno. El correo electrónico permite mandar o recibir información de manera asíncrona, gratuita y prácticamente instantánea.

Los medios de comunicación han perdido protagonismo, porque cualquier persona puede divulgar información en su blog o mediante su cuenta de Twitter. Youtube se ha convertido en uno de los canales de comunicación más influyentes y constituye una ventana al mundo para muchas personas que quieren darse a conocer como grupos musicales, actores, blogueros, etc. Muchos periódicos en papel se han visto en una situación económica delicada, en buena parte porque a través de Internet es posible conocer información gratuita y actualizada.

Muchas empresas ofrecen servicios únicamente a través de Internet como: servidores de dominios, gestión de publicidad *online*, servicios de *hosting*, *Cloud Computing*, venta de productos financieros, etc. Algunas de las empresas que vislumbraron el potencial de la Red han conseguido formar parte de las organizaciones más influyentes en la sociedad.

Muchos de los negocios y plataformas que se han ido desarrollando por este canal se han expandido a una velocidad de vértigo. Existen portales que han incrementado el número de usuarios de manera exponencial como Google, Twitter o Facebook. Para darse cuenta de este fenómeno, basta considerar que Google nació en 1996 y que el promedio de visitas en 2012 fue superior a 5.000 millones al día. El caso de Facebook también es bastante llamativo, nació el 4 de febrero del 2004 y en 2015 llegó a 1.100 millones de usuarios activos registrados.

1.1.2. Importancia del posicionamiento web en los buscadores

A pesar de que muchos expertos aseguran conocer cómo funciona el algoritmo de búsqueda de Google y de que explican cuáles son los factores más determinantes, este codiciado algoritmo es uno de los secretos mejor guardados y solamente es conocido por unos pocos trabajadores de esta compañía.

También es cierto que la experiencia compartida de miles de profesionales dedicados a la optimización en buscadores (*Search engine optimization*) (SEO) intentando posicionar páginas en los primeros resultados de búsqueda junto con los documentos realizados por personas en este buscador, dan bastantes pistas de qué es lo que más se valora a la hora de posicionar un artículo o una página.

Otra de las estrategias que muchos internautas han intentado llevar a la práctica con poco éxito consiste en intentar hacer una página web de tanta calidad que sean los mismos usuarios los que promocionan las páginas a través de foros y de redes sociales, atrayendo a más visitantes que repitan este mismo proceso.

Respecto a este punto, se ha de tener en cuenta que esto es algo complejo pues ya existen

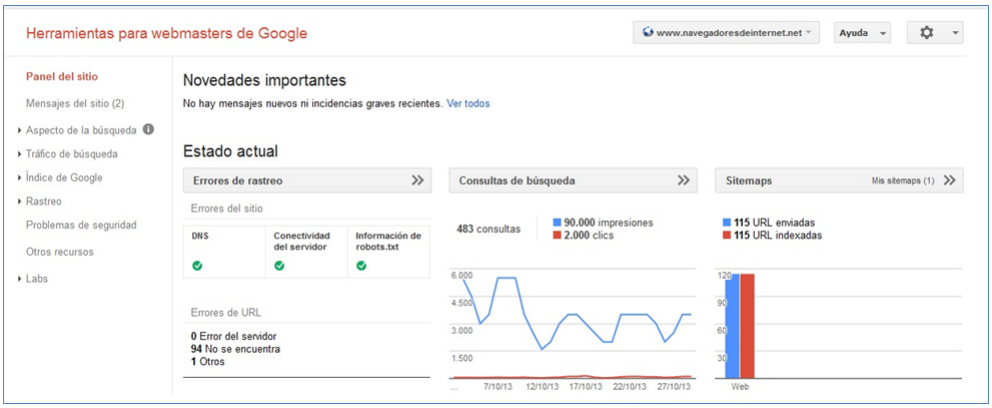

Figura 1.1: Herramienta para *Webmasters* de Google.

páginas con miles de artículos, con equipos de personas trabajando para ellas, con años de buena reputación en la Red y que ya constituyen un negocio en marcha.

Una estrategia menos ambiciosa pero más realista, y con más posibilidades de éxito, consiste en centrarse en un tema muy específico y en ofrecer la información de mayor calidad y más actualizada. Por ejemplo, uno podría centrarse en un equipo de fútbol en lugar de centrarse en el fútbol en general, o centrarse en un tipo de mascota más que en el mundo de las mascotas.

En este sentido, es muy importante poner en práctica las recomendaciones que vienen en la guía de Google de recomendación de diseño de páginas web. También es muy importante intentar que páginas web con un alto valor de página (*PageRank*) enlacen dicha página.

Al diseñar una página web hay que tener en cuenta dos conceptos fundamentales: navegabilidad y usabilidad. La navegabilidad es la facilidad con la que un usuario puede encontrar la información en un portal. La usabilidad hace referencia a la facilidad que tiene una herramienta para que un usuario pueda realizar su propósito. Google ha creado la herramienta *Webmaster tools* para mejorar la calidad del usuario. En la Figura 1.1 se muestra una imagen de dicha herramienta.

El posicionamiento web consiste en aparecer en las búsquedas orgánicas de los motores de búsqueda (*Search Engine Results Pages*) (SERP) que son las páginas ordenadas como respuesta de una búsqueda. Para lograr un buen posicionamiento web existen técnicas aceptadas por los buscadores que se denominan *White-hat* las cuales son éticas, y también hay otras técnicas conocidas como *Black-hat* que intentan engañar a los buscadores para obtener más visitas.

Figura 1.2: Primer *banner* de la historia.

1.2. Contexto del problema

En los siguientes puntos se describe el marco de la tesis. La mayoría de los artículos de investigación existentes sobre publicidad *online* tratan principalmente sobre dos temas: estimar el valor del *Click-through rate* (CTR) y la detección de fraude. Estos son los principales desafíos de la publicidad *online* y sobre estos temas se publican cada año un gran número de artículos.

Sin embargo, esta investigación aborda un problema que, aunque no muy estudiado en la literatura, tiene una vital importancia. La tesis doctoral está centrada en el diseño de un modelo de intercambio de anuncios entre pequeñas redes de publicidad *online* con el fin de mejorar su rendimiento. Por ello, se describe la situación de los modelos de intercambio de anuncios, las ventajas de las grandes redes y unas directrices sobre cómo debe ser un modelo de intercambio de anuncios.

1.2.1. Historia de la publicidad en Internet

La compañía AT&T desarrolló el primer *banner* en Internet. Este fue publicado en Hotwired en 1994 y se muestra en la Figura 1.2. Pero este *banner* es muy parecido a la publicidad tradicional que se puede hacer en una revista o en un periódico. Sin haber desplazado totalmente a los medios tradicionales, Internet se configura ya como uno de los principales soportes publicitarios.

Aunque la publicidad *online* tiene algunas peculiaridades, su filosofía y su estructura no dista mucho de cualquier estrategia publicitaria convencional: buscar personas potencialmente interesadas en un servicio y dirigirles un mensaje atractivo con la finalidad de que lo adquieran.

En los comienzos de Internet no tenía sentido desarrollar la publicidad debido al poco volumen de usuarios que había. De hecho, la idea inicial es que no hubiera publicidad, pero a medida que el número de usuarios fue aumentando la idea de usar la Red como medio publicitario se fue consolidando.

Los roles implicados en la publicidad *online* son los cuatro roles tradicionales del proceso publicitario: el anunciante, la agencia de publicidad, el soporte o medio publicitario, y el destinatario del mensaje.

Algunas personas equivocadamente pensaron que sería suficiente con tener un portal web atractivo para generar ventas *online*. Lo cierto es que sin una buena publicidad o un buen posicionamiento web esta página no servirá de mucho.

Inicialmente la publicidad en Internet se valió de fórmulas tan llamativas como intrusivas, lo que ha provocado que la mayoría de las personas no tenga un concepto bien definido sobre qué es la publicidad en Internet. Muchos entienden este concepto como esas ventanas gigantes que parpadean y que se mueven de un lado a otro de la página, en las que el usuario tiene que encontrar el modo de cerrarlas para visualizar el contenido. Otros se imaginan esos *banners* que parpadeaban incesantemente dificultando la navegación con fotos de provocativas modelos.

Esto es una forma de publicidad de baja calidad, intrusiva y en vistas a desaparecer. Las principales formas de publicidad y las que más ingresos generan apenas llaman la atención, pero están presentes cada vez que se hace una consulta en un buscador, que se accede a una página web, o que se reproduce un vídeo.

De modo progresivo, la publicidad ha ido evolucionando hacia formas más discretas, sutiles y eficaces. Es el caso de los enlaces patrocinados que ofrecen los buscadores, los anuncios contextualizados que presenta el proveedor de correo electrónico, o los breves *spots* publicitarios en servicios como Youtube o Spotify. De todos los formatos de anuncios que permite la Red, los más frecuentes son los *banners*, seguidos de los enlaces patrocinados en los buscadores, y los vídeos.

Algunos ejemplos frecuentes de publicidad en este medio podrían ser:

- Cuando se teclea en `Google.com` las palabras clave: “Alquilar coche en Sevilla”. En los resultados de búsqueda que devuelve la consulta, normalmente, los dos o tres primeros que aparecen con fondo ligeramente sombreado son enlaces patrocinados.

- Al acceder a `Youtube.com` para visualizar un vídeo, antes de poder acceder al contenido, hay un pequeño vídeo de publicidad.

- Cuando las personas ven las fotos de sus amigos en `Facebook.com`, normalmente hay una serie de enlaces con imágenes ofreciendo algún producto o servicio. Facebook factura a estos anunciantes cada vez que un usuario hace clic en ellos o por cada mil veces que se imprime. Los anuncios de Facebook están orientados a cierto perfil y tienen en cuenta características de los usuarios como la edad, el género, la localización, los gustos o las aficiones.

La publicidad en Internet constituye la principal fuente de ingresos de los editores de páginas web. Los principales modelos de pago de estos editores son tres: *Cost-per-mille* (CPM) por el número de veces que se muestra un anuncio, *Cost-per-click* (CPC) por el número de clics

que un anuncio recibe, o *Cost-per-action* (CPA) por determinadas acciones que realicen los usuarios en la web del anunciante. Existen muchas estrategias fraudulentas, que se llevan a cabo por personas o por robots que intentan falsear estos datos para ganar más dinero a costa de los anunciantes.

A un anunciante le resulta difícil comprobar que sus *banners* se presentan en pantalla a determinados usuarios cuando entran en una determinada página web. Si bien es cierto que existe el *Return over investment* (ROI) que indica el beneficio sobre la inversión, como las grandes compañías se anuncian en radio, prensa, televisión e Internet a la vez, es muy difícil determinar el porcentaje que supone la publicidad *online*.

Con la finalidad de regular y armonizar la publicidad en Internet, la propia industria publicitaria creó en 1996 el *Interactive Advertising Bureau* (IAB). Desde su constitución, el IAB ha desarrollado una amplia tarea, estableciendo estándares de conducta, homologando los formatos publicitarios, y contribuyendo a combatir los fraudes y los abusos publicitarios en Internet. En 2014, la IAB englobaba a más de quinientos medios de comunicación y empresas de tecnología que representaban el 86 % de la publicidad *online* de los Estados Unidos [11].

1.2.2. Modelos de intercambio de anuncios

A lo largo de los años, la selección de un anuncio se ha convertido en una decisión crítica para las redes y para todo el ecosistema que las rodea. El proceso de selección de un anuncio en el Modelo de Intercambio de Anuncios (MIA) está compuesto por un conjunto de pasos presentados de forma secuencial, pero estos eventos son dependientes entre sí [1].

Los MIAs difieren en la forma de tratar estas dependencias. Por ejemplo, el evento "Selección de un anuncio" se trata en algunos métodos de selección como un módulo independiente, que tiene en cuenta la similitud literal o semántica entre las consultas y las palabras clave durante el proceso de selección de un anuncio [12]. Mientras que en otros métodos se describe como un proceso dependiente de las preferencias del usuario, la consulta en el buscador, el precio y otras variables.

Algunos autores clasifican el MIA en tres etapas principales: las impresiones publicitarias, la gestión de ingresos y la selección de un anuncio [6]. Sin embargo, otros autores solamente los clasifican en dos categorías. La primera categoría se basa en la importancia de las palabras clave, y calcula la puntuación según la relevancia. Y la segunda categoría se basa en la relación semántica entre las consultas y las palabras clave [12].

Otros estudios de investigación se enfocan en estimar la probabilidad de que se haga clic en un anuncio. Otros utilizan métricas para puntuar la calidad de los anuncios. En este caso, se asigna la probabilidad de que un usuario genere un clic en un anuncio y, posteriormente,

los anuncios se clasifican según el valor del CTR. El anuncio mejor clasificado tendrá más probabilidades de ser mostrado al usuario. Después de que un usuario haga clic en uno o más anuncios, el anunciante pagará el segundo precio más alto de la subasta [12, 13].

Otros métodos formalizan el proceso de selección de anuncios, como un problema de optimización combinado de control estocástico [6]. Estos métodos desarrollan una política para la asignación de publicidad *online* basada en la calidad del anuncio y en el rendimiento económico.

A estos modelos se añaden nuevas variables que aumentan la complejidad del MIA. En esta línea, algunos autores se centran principalmente en la optimización de ingresos [14]. Para ello, minimizan los costos de inventario total mediante la búsqueda del intervalo óptimo y la cantidad óptima del pedido.

Google introdujo en el mercado su sistema de *Pay-per-click* en 2002, desde entonces han surgido diversos estudios en relación con la publicidad *online*. Sin embargo, el precio por segunda subasta (*Global Second-Price*) (GSP) sigue siendo uno de los mecanismos de subasta más implementados. El GSP es un mecanismo de subasta en el que la puja más alta gana la subasta y el comprador paga el precio de la segunda cantidad más alta [12, 13].

Algunos investigadores recomiendan no utilizar el sistema tradicional de oferta sino una versión mejorada del mecanismo de subasta de Vickrey [15]. Este sistema mejora las probabilidades de que el anunciante no sufra un engaño en el precio que cobra a los anunciantes en el sistema de apuestas. Esto se logra mediante la creación de mecanismos transparentes y verificables [16].

Otra investigación indica que el actual mecanismo de fijación de un precio suelo en el sistema de apuestas en tiempo real coloque al anunciante en una posición desfavorable y que se deben desarrollar algoritmos que toman esto en cuenta. Muchos eventos tienen un patrón periódico como las impresiones, el número de clics, las ofertas y los tipos de conversión [2].

En este tipo de investigaciones se cambia la perspectiva de marketing y podría mejorar las expectativas de anunciantes y de los ingresos de manera significativa como se describe en [3]. En el proceso RTB, los anunciantes compiten entre sí para poder mostrar sus anuncios en el sitio web del editor. En un milisegundo se deben controlar las impresiones de aplicaciones móviles y las impresiones de páginas web que darán preferencia a un anunciante particular cuando se muestre un anuncio en una aplicación.

1.2.3. Ventajas de las grandes empresas

Las principales ventajas que tienen las grandes redes como Google o Facebook sobre las pequeñas redes son:

- Costes variables vs. costes fijos: En casi todos los productos hay dos tipos de costes,

que son los variables y los fijos. En la fabricación de coches, los costes fijos serían los costes invertidos en el desarrollo de un modelo como el diseño, los laboratorios, las pruebas, etc. Y los costes variables serían el dinero que supone cada uno de los coches como el material, las horas de trabajo de las personas, la energía necesaria, etc. Del costo total, un 40 % pueden ser gastos fijos y un 60 % pueden ser gastos variables.

En cambio, en los negocios de Internet casi todos los costes son fijos. Cuando se diseña un software los costes fijos son las horas de programación y de investigación. Y los costes variables son el valor del CD donde se almacena el programa. Por lo tanto, los fijos son de un 98 % y los variables de un 2 %.

- Volumen y margen: Cuando se lanza un producto al mercado se puede orientar a ganar dinero por margen o por volumen. Ferrari vende pocos coches, pero por cada coche tiene mucho margen. En cambio, Coca-Cola tiene poco margen, pero vende muchas botellas.

  Google ha entendido que el negocio en Internet no está en el margen sino en el volumen. Poca gente está dispuesta a pagar por un servicio en Internet, pero millones de personas están dispuestos a usarlo gratis y estos usuarios sirven para mostrarles anuncios y generar dinero. Por lo tanto, las empresas con pocos usuarios difícilmente son autosuficientes.

- Sinergias: Cuando una gran empresa desarrolla un módulo, este módulo se puede incorporar en muchos de los productos de esta empresa. Por ejemplo, gracias a que Google tiene un buscador puede usar ese motor de búsqueda e incorporarlo en Youtube, Gmail, etc. Gracias a que Google tiene Gmail y Adsense, puede publicitar Google Chrome en estas plataformas.

- Financiación: Estas empresas son tan poderosas y tienen unos ingresos tan altos que no tienen problemas de financiación. Cuando Google compró Youtube, pudo ofrecerlo sin ningún anuncio durante algunos años. Cuando tuvo el suficiente número de usuarios y se consolidó como líder indiscutible en el mercado, decidió hacer publicidad.

- Resistencia al cambio: Una vez que un usuario se acostumbra a usar una plataforma, o tiene un correo electrónico en un servicio, para que dicho usuario cambie, tiene que superar una barrera de adaptación. La principal ventaja de Google es que tiene casi todo el mercado y que tiene un público cautivo.

- Seguridad frente al fraude: Uno de los principios de Kerckhoffs [17] es que el éxito de un algoritmo criptográfico no debe residir en que permanezca secreto. Cualquier

algoritmo que se emplea en criptografía es publicado y si el sistema es susceptible de ser atacado de manera eficaz, automáticamente se mejora o deja de utilizarse. Esta política ha permitido que los algoritmos sean cada vez más seguros y que a día de hoy sean casi invulnerables.

Sin embargo, en la publicidad en internet la realidad es muy diferente. Uno de los motivos por los que la gente confía en Google es por su capacidad de detectar clics fraudulentos. Si Google publicara sus algoritmos de detección de clics, todas las plataformas serían igual de seguras y esto ya no supondría una ventaja competitiva. Con lo que cada plataforma tiene que idear sus propios sistemas.

1.3. Objetivos de la tesis

Las grandes redes como Google o Facebook están acaparando cada vez más, un mayor porcentaje en la publicidad en Internet. Esto constituye un gran peligro pues dificulta que puedan desarrollarse nuevas redes publicitarias. En esta tesis se pretende realizar un modelo de intercambio de anuncios en el que las pequeñas redes puedan unirse para beneficiarse mutuamente. De forma que puedan aumentar el rendimiento de sus campañas y mejorar la detección de fraude. Este modelo se desarrolla bajo una nueva perspectiva que consiste básicamente en incluir los criterios necesarios para que el ecosistema publicitario funcione de manera correcta al seleccionar un anuncio.

Se considera que algunas técnicas emergentes de la Inteligencia Artificial como el aprendizaje automático (*Machine Learning*) (ML), los algoritmos genéticos y los métodos de selección de variables puedan hacer esto más viable.

Para poder llevar a cabo dicho modelo se definen los siguientes objetivos:

1. Analizar el estado del arte del intercambio de anuncios y sus principales aspectos relacionados: En esta tesis se pretende desarrollar un modelo de intercambio de anuncios con un nuevo enfoque. Antes de desarrollar dicho modelo se deben estudiar a fondo las principales investigaciones relacionadas con los modelos de intercambio de anuncios. También se han de estudiar los principales aspectos relacionados con la publicidad en internet como son los navegadores, los buscadores, las ventajas de este tipo de publicidad, los modelos de pago, los roles que intervienen, el fraude en la publicidad y los formatos de anuncios. Por otra parte, se deben considerar otras disciplinas que juegan un papel fundamental con este tipo de publicidad como son el aprendizaje automático, los métodos supervisados de clasificación y regresión, los métodos de selección de variables, la computación evolutiva y los algoritmos genéticos.

2. Evaluar la viabilidad de la colaboración entre redes: En primer lugar, se ha comprobar si el rendimiento de las redes cuando estas intercambian anuncios e información se incrementa. Aunque esto a primera vista pueda parecer obvio, se han de hacer varias simulaciones para ver como aumenta el rendimiento en función del número de redes que participan. El margen de mejora está en función de varios factores como son el número de redes, el volumen de anuncios de estas, el número de anunciantes, el tipo de campañas de los anunciantes y la información que intercambian sobre el fraude.

   Además de comprobar que la colaboración en sí es positiva, es necesario crear un conjunto de algoritmos que permitan intercambiar anuncios de manera eficaz y en un tiempo reducido. Por lo tanto, es necesario comparar varios algoritmos de intercambio de anuncios y evaluar su rendimiento.

3. Definir los criterios que debe cumplir un modelo de intercambio de anuncios entre pequeñas redes adecuado: Este nuevo modelo, bien podría ser una semilla que permitiera a editores y a anunciantes disfrutar de un modelo en el que todos se vean beneficiados. Los criterios que se han considerado son maximizar los ingresos económicos, procurar una distribución equitativa entre las redes de anuncios que participan, conseguir que todos los anunciantes puedan mostrar sus anuncios, velar por la rentabilidad de las campañas de los anunciantes y evitar el fraude en la publicidad.

4. Diseñar una metodología para optimizar el modelo de colaboración entre redes: De nada sirve crear un modelo que no sea eficiente, por lo que además de desarrollar un modelo de intercambio de anuncios, se debe desarrollar una metodología que sea fácilmente adaptada y optimizada. Para dicho fin, la metodología desarrollada consta de varios pasos. En primer lugar se definen los objetivos que debe tener un ecosistema publicitario para funcionar de manera adecuada. Después, se definen una serie de penalizaciones para garantizar que los objetivos se cumplan. Es decir, que cuando un objetivo no se cumpla acarreará una penalización en forma de sanción económica. Además, como el fraude es uno de los aspectos más preocupantes se debe desarrollar una política antifraude, así como unas reglas para expulsar a quien comenta trampas. Posteriormente, se define una fórmula en la que se representan cada uno de los objetivos y en donde a cada objetivo se le asigna un peso. Finalmente, se optimizan los pesos mediante un algoritmo genético que prueba muchas combinaciones y que selecciona la que tiene mejor *fitness*. El *fitness* responde a una métrica establecida que consiste en restar a los ingresos obtenidos por el modelo el sumatorio de todas las penalizaciones.

5. Analizar, evaluar y testear técnicas de aprendizaje automático para publicidad online:

El aprendizaje automático ha supuesto una gran revolución en muchos campos de la ciencia pues permite predecir el comportamiento de muchos modelos de forma automática. Es decir, crea un modelo a partir de un conjunto de muestras con el fin de que el modelo sea capaz de predecir la salida de futuras muestras.

Aunque las técnicas de aprendizaje automático son muy útiles y permiten obtener un gran rendimiento en poco tiempo, también es cierto que requieren una serie de conocimientos previos. Entre ellos, se debe hacer un buen preprocesamiento de los datos y se deben seleccionar muestras representativas y con garantías de calidad. También se deben aplicar métodos de selección de variables, ya sean de tipo *wrapper*, de tipo filtro o de ambos tipos, para crear modelos más sencillos, más rápidos y con mejores resultados. Es importante encontrar aquellos métodos de aprendizaje automático que mejor se adapten al tipo de problema que se está resolviendo. En esta tesis vamos a evaluar un método supervisado relativamente reciente llamado *Deep Learning* y un método de selección de variables muy novedoso llamado ENORA.

El algoritmo ENORA de selección de variables ENORA no ha sido desarrollado en esta tesis, pero debido a que ha sido creado por uno de los directores de la misma parece interesante evaluar su rendimiento en el dominio de la publicidad *online*. La selección de variables puede verse como una paso intrínseco en el aprendizaje automático. La selección de variables es un paso previo a la construcción de los modelos. Este proceso es muy útil pues permite eliminar las variables redundantes y las que no aporten información. Trabajar con menos variables permite hacer modelos más precisos, de menor tamaño y en un tiempo menor. También mejora el tiempo de respuesta pues analiza menos variables. Por todos estos motivos se compara el algoritmo de tipo *wrapper* ENORA con otros algoritmos muy utilizados como son el RFE y el NSGA-II. ENORA se evaluará tanto para regresión en la predicción de ventas *online* como para clasificación en la estimación del CTR.

6. Establecer mecanismos para el cálculo del valor de un anuncio en los modelos CPM, CPC y CPA: En la publicidad en internet se aplican muchos métodos de pago. Los métodos más extendidos son el CPC, seguido del CPM y del CPA. Algunos anunciantes prefieren un método de pago a otro. Para que en una misma red publicitaria puedan coexistir los tres modelos de pago, es necesario desarrollar una metodología que calcule el valor de un anuncio en términos económicos.

   Para calcular el valor de un anuncio en los tres principales métodos de pago se deberían crear tres módulos: uno que prediga la probabilidad de que el anuncio sea *spam*, otro que prediga la probabilidad de que se genere un clic y otro que estime la pro-

babilidad de venta a partir de un clic. Para predecir el valor de un anuncio en estas tres modalidades se va a diseñar un sistema compuesto de los módulos anteriores que estime el valor de un anuncio a partir del modelo de pago y ciertas características del editor y el usuario. Algunas características son el tamaño del *banner*, el dispositivo del usuario, la temática de la página web, o el navegador del usuario.

1.4. Estructura del documento

Una vez que la tesis ha sido motivada y se han marcados los objetivos, el documento se ha organizado con la siguiente estructura:

- Capítulo 2 - Fundamentos: En los fundamentos se hace un análisis profundo sobre la publicidad en Internet y su situación actual. Dentro de la publicidad se habla de los formatos de anuncio y de los roles que participan en el proceso publicitario. También se habla de los métodos de pago y de los tipos de fraude que se presentan en cada método de pago.

  Se hace un repaso de la situación de los modelos de intercambio de anuncios actuales y de cómo se calcula el valor de un anuncio en los distintos modelos de pago. También, se da un breve repaso sobre algunas de las disciplinas que se utilizan para desarrollar este modelo de intercambio de anuncios como son el ML, la selección de variables y los algoritmos genéticos.

- Capítulo 3 - Viabilidad de la colaboración entre redes para aumentar el rendimiento económico y mejorar la detección de fraude: En este capítulo se muestra cómo mejora el rendimiento de las redes cuando colaboran, tanto en el aspecto económico como en la detección de fraude. También se comparan varios algoritmos de distribución de visitas para que todas las redes puedan recibir anuncios. Para ello, se crean algoritmos para repartir las visitas entre las distintas redes en el menor tiempo posible.

  Para aumentar la cobertura, que se define como el porcentaje de anuncios que consiguen ser mostrados sobre el total de anuncios, se aplican matrices de similitud. Para optimizar el intercambio de anuncios entre redes se prueban varios algoritmos para reducir al máximo el número de comparaciones y el tiempo requerido para asignar un anuncio a un espacio pues este tiempo debe ser menor a un umbral. Para ello se aplican los conceptos de árboles AVL y Hadoop.

- Capítulo 4 - Diseño de un modelo multicriterio de intercambio de anuncios y su optimización mediante un algoritmo genético: En este tema se propone diseñar un

Modelo de Intercambio de Anuncios (MIA) entre las pequeñas redes publicitarias como una solución para evitar su desaparición y para impulsar la creación de nuevas redes publicitarias.

Este modelo tiene en cuenta un conjunto de objetivos que se centran en: obtener el máximo rendimiento del modelo, garantizar la satisfacción de los usuarios, los anunciantes y las redes de publicidad, y evitar el fraude mediante una serie de reglas y de penalizaciones. Cada uno de los objetivos tiene asociado un peso cuyo valor se optimizan mediante un algoritmo genético.

- Capítulo 5 - Diseño de los módulos implicados en el cálculo del valor de un anuncio mediante métodos supervisados de *Machine Learning*: A lo largo de este capítulo se calcula la rentabilidad de un anuncio para los modelos CPM, CPC y CPA. Para calcular de manera eficiente el valor de un anuncio se ha calculado la probabilidad de que sea *spam*, la estimación del CTR y la probabilidad de generar una venta mediante métodos supervisados de aprendizaje automático.

  Se ha diseñado un módulo para cada uno de las probabilidades y se han relacionado para componer un sistema que determine el valor del anuncio en función del modelo seleccionado por el anunciante. Este modelo tiene en cuenta un conjunto de objetivos que se centran en: obtener el máximo rendimiento del modelo, garantizar la satisfacción de editores, anunciantes y redes de publicidad, y evitar el fraude mediante una serie de reglas y de penalizaciones. Cada uno de los objetivos tiene asociado un peso cuyo valor se optimiza mediante un algoritmo genético.

- Capítulo 6 - Optimización de los módulos mediante ENORA y otros métodos de selección de variables: Un aspecto fundamental para mejorar el rendimiento del modelo de intercambio de anuncios es la selección de los atributos que forman parte de los datos tenidos en cuenta para la predicción de ventas y la estimación del CTR. En este capítulo se aplican distintos métodos de selección de variables y los resultados han sido comparados utilizando diversas métricas (ACC, AUC, RMSE, OOB, tiempo de entrenamiento, tamaño del modelo) así como el test estadístico *Paired t-tets* corregido.

  Para la predicción de ventas, los modelos generados son de regresión, y se han utilizado los métodos de tipo *wrapper* ENORA, NSGA-II y RFE usando *Random Forest* como método de regresión. Para la estimación del CTR se han utilizado modelos de clasificación, y se han aplicado métodos de selección de variables tanto de tipo filtro (PCA e Information *Gain Ratio*) como de tipo *wrapper* (ENORA y NSGA-II con *Random Forest*).

- Capítulo 7 - Conclusiones y trabajos futuros: En el último capítulo se muestra las conclusiones, las contribuciones, los trabajos futuros de la tesis doctoral y un elenco de las publicaciones derivadas y de las relacionadas con la tesis doctoral.

Capítulo 2

Fundamentos

2.1. Introducción

Desde la publicación del primer *banner* en 1994 hasta el día de hoy, la publicidad en Internet ha tenido un enorme crecimiento [18]. Yahoo[1] (1994), Google (1998) y Facebook (2004) se han centrado en la publicidad *online* y en pocos años han llegado a figurar entre las empresas más importantes del mundo.

Algo similar ocurre con Amazon (1994) y Ebay (1995) que son empresas especializadas en vender por Internet. Cada vez los usuarios compran más a través de Internet pues pueden encontrar casi cualquier producto a menor costo y con la comodidad de poder comprar desde casa [19].

Las redes publicitarias necesitan equipos especializados [20] para mejorar permanentemente los algoritmos relacionados con el proceso de publicidad *online* tales como la detección de fraude [21] y la optimización del proceso de selección del anuncio más rentable.

Las redes publicitarias rara vez publican información acerca de los algoritmos involucrados en la gestión de publicidad porque las investigaciones para el desarrollo de estos algoritmos requieren grandes recursos financieros y representa una ventaja competitiva respecto a sus competidores. A pesar de ello, el interés general en esta área es tan grande que cada año se publican cientos de artículos científicos.

El número de ventas *online* en la última década ha aumentado año tras año. Y las previsiones indican que este número seguirá aumentando. Las personas de edad avanzada son más reacias a la hora de comprar *online*, pues muchos de ellos tienen dificultad en el uso de las nuevas tecnologías. También han adquirido una serie de hábitos que son difíciles de cambiar.

Por el contrario, las personas jóvenes son más propensas a comprar a través de Internet

[1] El número entre paréntesis hace referencia al año en el que la compañía fue fundada.

y cuando ven un producto a un buen precio suelen compartirlo con sus amigos a través de las redes sociales. Las compras a través de Internet tienen grandes ventajas para los clientes pues se evitan tener que viajar y hacer colas. El producto puede ser adquirido casi instantáneamente.

Internet permite a los usuarios comparar el precio de los productos rápidamente y por consiguiente, los precios de los productos están más ajustados, lo que supone otro beneficio para los usuarios. Mientras tanto, el vendedor tiene la ventaja de ser capaz de vender sin tener que pagar alquiler o tener que contratar a dependientes para la tienda [22].

Además, el vendedor puede almacenar los datos del usuario a través de una página web. Entre los datos que pueden ser almacenados está el navegador, el tiempo de acceso, el sistema operativo, el tipo de dispositivo, el tiempo empleado en el sitio, las zonas de una página web más transitadas, y un largo etcétera. A veces los productos vendidos a través de la red no son tangibles. Hoy en día, es posible vender la información sin el correspondiente soporte físico. Vender un libro o un disco consiste en la descarga de un documento electrónico en un determinado formato.

Cada vez más empresas anuncian sus productos en la red porque es el canal más adecuado para llegar a los usuarios. Los productos intangibles son muy apropiados para ser vendidos a través de este canal pues no requieren de los sentidos humanos. Algunos ejemplos de este tipo de productos son las reservas de hoteles, los billetes de avión, las acciones de una compañía o los seguros médicos.

Por este canal las empresas son capaces de vender a lo largo de todo el mundo, por lo que sus clientes potenciales se cuentan en miles de millones. En 2016, el número de usuarios de Internet superaba los tres mil millones [23] y todas las previsiones indican que este número seguirá aumentando [24].

La publicidad *online* ofrece a los anunciantes grandes ventajas a la hora de orientar sus campañas hacia un público específico o de hacer modificaciones en tiempo real. Esto explica por qué cada vez más anunciantes eligen hacer sus campañas a través de Internet. Las redes de anuncios permiten a los anunciantes publicar sus anuncios en las páginas de los editores.

Los navegadores y los buscadores han permitido a los usuarios acceder a internet de manera fácil y amigable. Gracias al desarrollo de estos, no se necesita ser un experto para poder navegar por la red. La publicidad *online* tiene ventajas muy significativas respecto a la publicidad tradicional que explican su crecimiento ininterrumpido desde su aparición.

2.1.1. Nacimiento de los navegadores

Los navegadores permiten que personas sin conocimientos avanzados de informática, puedan circular por la Red de forma sencilla. Los enlaces que conforman los menús de las

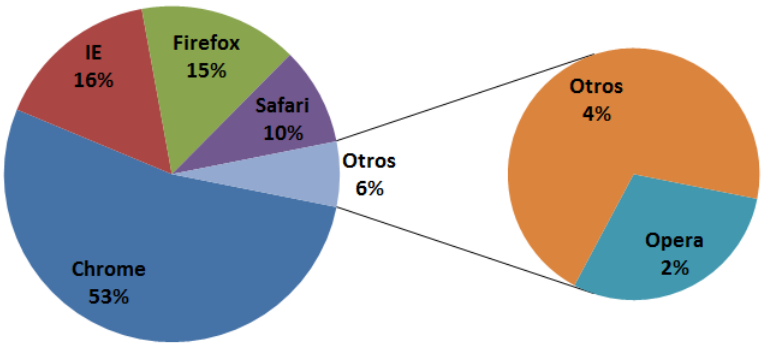

Figura 2.1: Estadísticas del uso de navegadores (Jul. 2015 - Dic. 2015).

páginas permiten a los usuarios seleccionar la información que les interesa. Los navegadores permiten retroceder a un punto anterior si la información a la que se accede no es del interés de los usuarios.

El primer navegador fue desarrollado en 1990 por Tim Berners-Lee. Y como solamente funcionaba en estaciones Next no se hizo muy popular. El navegador Mosaic tuvo mayor aceptación, se desarrolló tanto para los SO Windows como para Apple. Netscape tuvo una expansión muy rápida pero no duró mucho porque Microsoft lanzó una versión de Mosaic que actualmente se conoce por Internet Explorer. Este navegador venía por defecto en el sistema operativo Windows, por lo que la mayor parte de la gente lo usaba.

Netscape publicó el código fuente del navegador dando lugar al proyecto Mozilla, pero al estar desarrollado en un lenguaje un tanto obsoleto se tuvo que volver a programar. En 2002 se lanzó al mercado una versión de Mozilla muy eficiente basada en un lenguaje de programación más avanzado. Mozilla Firefox y Google Chrome aparecieron en los años 2004 y 2008 respectivamente.

A pesar de que existen navegadores con mucha calidad como Opera, Safari o Apple, sin duda alguna, Google Chrome es el que ha alcanzado mayores cuotas de mercado como puede verse en la Figura 2.1 [25]. Este navegador se ha convertido en el más utilizado por los internautas, principalmente por la gran promoción que ha hecho Google. Cuando se accede a las páginas `Google.com`, `Gmail.com` o `Youtube.com`, estas invitan continuamente a instalar el navegador Google Chrome como se puede apreciar en la Figura 2.2.

2.1.2. Los primeros buscadores

Antes de que se desarrollaran los buscadores se usaban los directorios que son listas de páginas web agrupadas por categorías y subcategorías. Normalmente, los directorios tienen

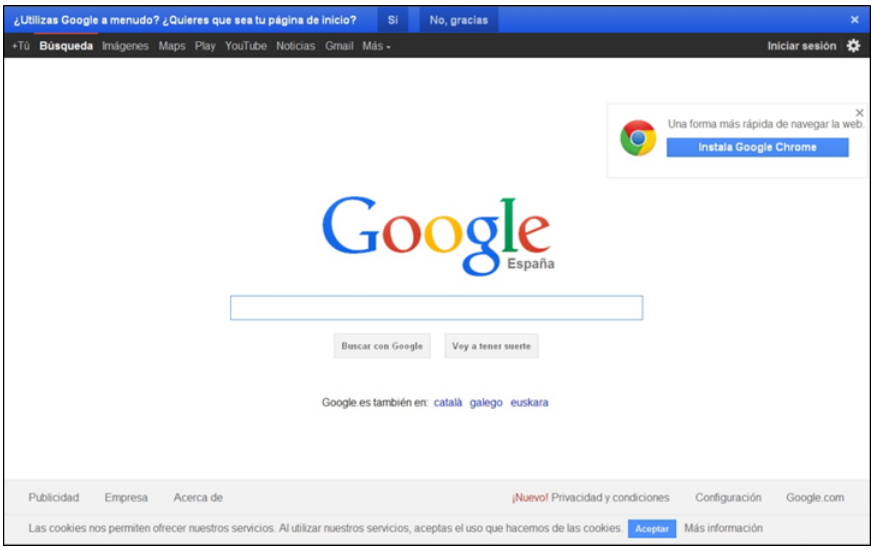

Figura 2.2: Google recomienda usar el navegador Chrome (12/11/2015).

campos que dan información sobre la página, entre los metadatos más frecuentes se encuentran: título de la página, descripción de la página, palabras clave, una pequeña puntuación y un enlace a la página. Los directorios tuvieron algo de popularidad en los principios de Internet pero fueron superados rápidamente por los buscadores gracias a su facilidad de uso y su rapidez.

Un buscador o motor de búsqueda es una aplicación web que muestra a los usuarios las páginas mejor relacionadas con las palabras clave que se introdujeron. Internamente los buscadores utilizan un *spider*, que es un programa que descubre las nuevas páginas a partir de los nuevos enlaces, para rastrear las nuevas páginas que van creando los usuarios. Se puede considerar como antecesores de los buscadores dos herramientas llamadas Archie y Gopher, que servían para buscar documentos dentro de los servidores FTP.

El protocolo HTTP establece la semántica para que servidores y clientes puedan comunicarse a través de la Red. Los usuarios hacen peticiones a los servidores a través de los navegadores que interpretan el protocolo HTTP para mostrar a los usuarios las páginas de una forma más amigable.

HTTP no tiene estados y para almacenar información hace uso de las famosas *cookies*, que son pequeños archivos que guardan información de las sesiones, la hora de acceso y otros datos de los usuarios. Para identificar una página se hace uso de una dirección URL, que tiene el nombre del dominio de la página que se asocia con un servidor y una cadena alfanumérica para identificar la sección dentro de este dominio.

A medida que se iban incrementando el número de usuarios de Internet y el número de páginas web, encontrar información se convirtió en una tarea realmente tediosa. Por ello, se desarrollaron los buscadores. Wandex fue el primero en 1993, pero tuvo problemas cuando

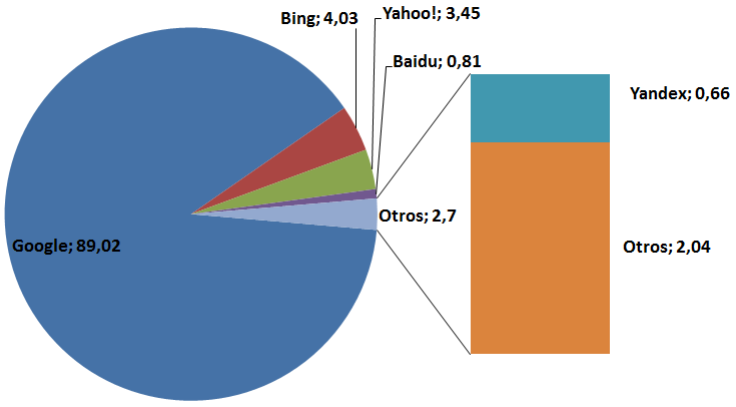

Figura 2.3: Estadísticas del uso de navegadores (Jul. 2015 - Dic. 2015).

se incrementaron las consultas de usuarios. Web Crawler apareció un año después y era mucho más sofisticado, más rápido y sus búsquedas tenían en cuenta los contenidos, no solamente el título o la URL, lo cual no hacían los buscadores desarrollados hasta esa fecha.

Lycos indexaba muchos documentos y tuvo una gran importancia. Pasó de indexar cerca de 400.000 páginas en 1996 a indexar 60 millones en 2005. En 1995 se desarrolló Excite y también aparecieron Yahoo, Altavista y Ozú. Estos buscadores eran más rápidos y ofrecían a los usuarios mejores resultados que sus predecesores.

Por esta época también apareció Hotbot que no terminó de despegar por una serie de inconvenientes y posteriormente fue comprado por Yahoo. Ask se propuso responder a las preguntas que los usuarios pudieran realizar a través del navegador y alcanzó cierta fama, a pesar de no conseguir su ambicioso objetivo.

El algoritmo de búsqueda de Google, el mejor posicionado actualmente, fue desarrollado por Sergey Brin y Larry Page, de la Universidad de Stanford en 1996. Éste tuvo una gran acogida por la precisión de sus resultados, su rapidez y su simplicidad. El algoritmo se basaba no solamente en la calidad de las páginas, sino que aplicaba un concepto bastante novedoso llamado *PageRank* [2], que tiene en cuenta la reputación de la página en la Red mediante los enlaces de otras páginas hacia esta.

Google se ha constituido como el líder de los buscadores alcanzando en 2013, unas cuotas de mercado de casi el 90 %, mientras que Yahoo tiene el 4 % y Bing el 3 %, los demás se reparten el 3 % restante como puede verse en la Figura 2.3.

[2]El *PageRank* establece la importancia de una página de forma numérica. El *PageRank* es un valor de 0 a 10, asignado por el algoritmo del buscador de Google a una página. Los usuarios solamente pueden ver el valor de tipo entero aunque internamente Google si maneja decimales.

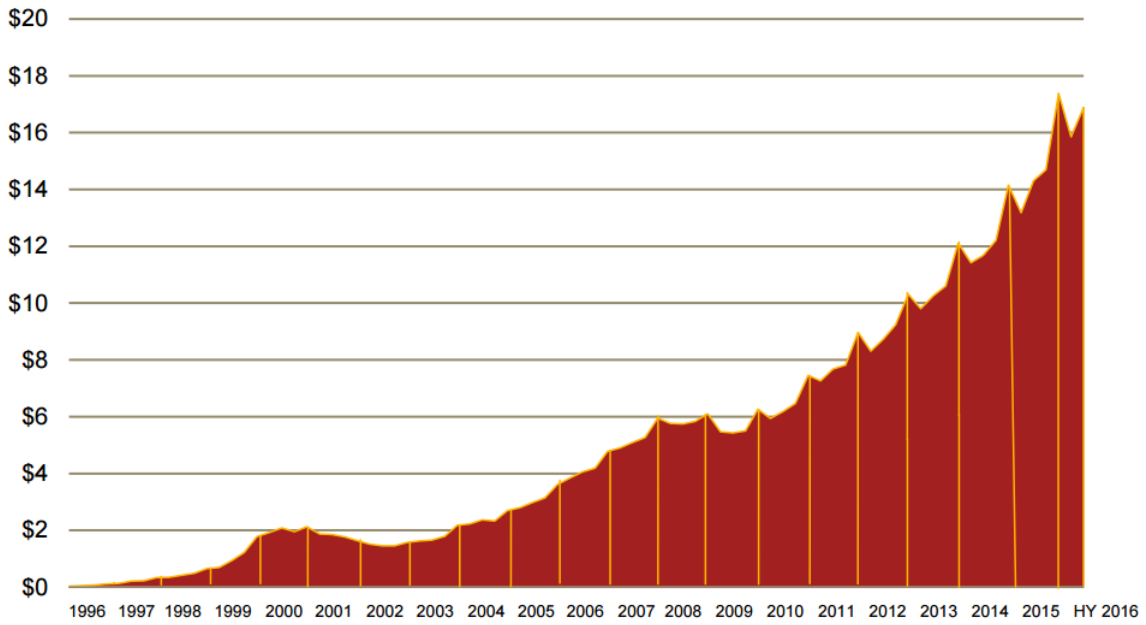

Figura 2.4: Ingresos en publicidad online en miles de millones de $.

2.1.3. Ventajas de la publicidad en Internet

Hay buenas razones que explican por qué la publicidad *online* ha tenido un crecimiento tan grande desde su creación en octubre de 1994 hasta la fecha actual como se muestra en la Figura 2.4 [26]. La publicidad en Internet tiene ventajas muy significativas respecto a la publicidad tradicional, lo que explica su gran aceptación por parte de los anunciantes y su constante crecimiento.

Las principales ventajas que ofrece este tipo de publicidad son:

1. Interacción con el público: Se establece un canal bidireccional. Los destinatarios no son meros receptores del mensaje, ya que pueden interactuar con la publicidad visitando la página, haciendo clic en “Me gusta”, compartiendo en Google+ o haciendo comentarios.

   La publicidad tradicional consiste básicamente en impactar al público objetivo para persuadirlo a comprar un producto. Algunas de las formas más comunes de interacción por parte de los internautas son visitar la página del anunciante, recomendarla, dejar un comentario, hacer clic en “Me gusta”, rellenar un formulario o difundir la campaña a través de un perfil en una red social. Internet permite modificar la campaña en cualquier momento para optimizar los resultados.

   La mayoría de las plataformas de publicidad en la Red permiten hacer ajustes inmediatos en la campaña como: cerrarla, cambiar las zonas geográficas o el huso horario,

y aumentar o disminuir el número de anuncios. Todas estas opciones son impensables en otros medios como prensa, radio o televisión.

2. Modificación de la campaña en tiempo real: Las plataformas permiten hacer ajustes inmediatos en las campañas como cancelarlas, cambiar las zonas geográficas o aumentar el número de anuncios en una determinada franja horaria.

3. Accesible a cualquier presupuesto: Pueden lanzarse tanto campañas masivas y millonarias por grandes compañías, como modestas y acotadas por pequeños empresarios. Internet se ajusta a cualquier presupuesto, no requiere necesariamente de inversiones publicitarias muy fuertes.

4. Segmentación de mercado: Permite clasificar muy bien a los internautas y dirigir los anuncios a grupos muy específicos [27]. Internet permite segmentar de manera muy precisa a los receptores de los mensajes. En su navegación, los usuarios dejan mucha información sobre sus necesidades, hábitos de consumo y preferencias.

   Esta información permite clasificar a los internautas de forma concisa y dirigir los anuncios a grupos muy específicos de personas que pueden estar interesadas en el servicio ofrecido por el anunciante. Esto es lo que se conoce comúnmente en marketing como *microtargeting* [28] y mejora las ganancias de los anuncios mediante una mayor aceptación por parte de los usuarios de los productos. El *Microtargeting*[3] se ha vuelto muy popular porque los anunciantes han notado que el éxito de sus campañas depende en gran medida de los parámetros de optimización [29].

   Si además, este público tiene un perfil en la plataforma como Facebook o de Google+, la información que se puede manejar es mucho más amplia como amigos, fecha de nacimiento, gustos, aficiones, eventos, páginas visitadas, etc. Otros parámetros que estas plataformas manejan sobre los usuarios son las palabras clave que se han tecleado, la dirección IP, su navegador, su conexión a Internet, su área geográfica y el tiempo de sesión.

5. Medición *in situ* de la efectividad: A través de distintas herramientas es posible medir el rendimiento de la campaña en tiempo real. Internet permite medir de forma concreta la efectividad de una campaña.

   Los anunciantes conocen el rendimiento de sus campañas en tiempo real, lo cual, permite ajustar parámetros de las campañas con el fin de hacerlas cada vez más eficaces

[3]El *microtargeting* consiste en dirigirse únicamente al segmento que se considera más apropiado para este producto. Si se anuncia una colonia para chicas jóvenes, las plataformas publicitarias de la Red permitirán segmentar por edad, género, aficiones, nivel de estudios, ciudad, horario, etc [28].

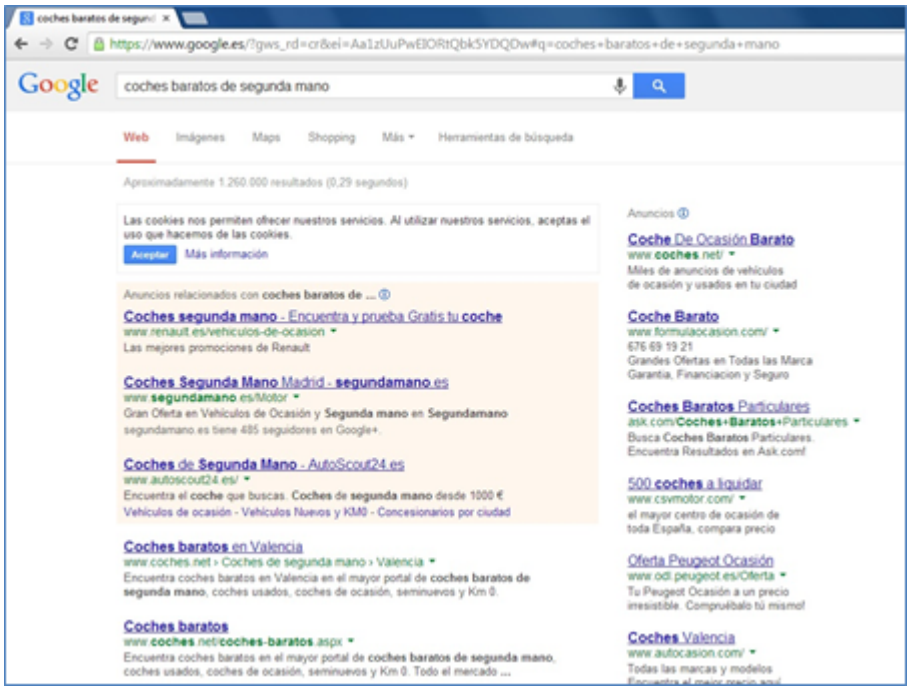

Figura 2.5: Los tres resultados sombreados son de pago por clic.

[30]. A través de las distintas herramientas es posible conocer cuántas personas han hecho clic en un determinado anuncio, cuánto tiempo han permanecido en la página web del anunciante, si han contratado algún servicio o si han vuelto a visitar la página en otra ocasión. Toda esta información permite al anunciante potenciar los aspectos positivos de su campaña, así como minimizar aquellos que no resultan rentables.

2.1.4. Formato de anuncios

Como se ha comentado anteriormente, el IAB ha fijado una serie de estándares para anunciarse en Internet, entre los que destacan:

- Anuncios de texto con enlaces patrocinados: Tienen la ventaja de que no llaman tanto la atención, por lo que si los usuarios generan clics es porque probablemente les interesa el producto. Algunas de las plataformas más importantes que trabajan con este tipo de anuncios son Google Adwords, Yahoo, Bing y Ask que suele incluir este tipo de anuncios en los enlaces patrocinados de los resultados de búsqueda.

  La diferencia entre los resultados orgánicos y los resultados patrocinados es que unos están ligeramente sombreados por lo que muchos usuarios no son capaces de distinguirlos. Como se puede ver en la Figura 2.5 los resultados sombreados responden a los resultados de pago.

Figura 2.6: Enlaces en los textos de la compañía Infolinks (18/2/2014).

- Enlaces en los textos: Hay varias compañías como Infolinks, Hotwords o Text-link, cuya publicidad consiste en subrayar algunas de las palabras y al pasar el ratón suele abrirse una animación como se muestra en la Figura 2.6. Si el usuario está interesado hará clic para dirigirse a la web del anunciante.

- Anuncios gráficos estándar: Los anuncios con imágenes también llamados *banners* son más llamativos y permiten reconocer la marca más rápidamente de manera visual. Los formatos más extendidos son el JPG y el GIF, también es muy frecuente el *flash* para hacer animaciones.

  Los banners son imágenes o secuencias de imágenes de distintos tamaños que promocionan distintos productos. Los tamaños más frecuentes son:

  - *Banner* (468×60): Éste es uno de los formatos más extendidos.
  - *Megabanner* (728×90): Debido a que las pantallas son cada vez más grandes los *banners* también aumentaron su tamaño.
  - Botón (150×60): Este tipo de formato suele tener un CTR más bajo, debido a que es un espacio muy pequeño.
  - Rascacielos (120×600): Este tipo de formato suele encontrarse a los lados de la página.
  - Robapáginas (300×250): Este formato tiene la ventaja de que ocupa mucho espacio por lo que suele tener un CTR muy alto.

- Anuncios gráficos especiales: Además de los *banners*, existen otras variedades de anuncios con distintas características. Para diseñar este tipo de anuncios suelen emplearse vídeos, sonido y otros efectos. La tecnología *flash* permite diseñar anuncios muy creativos, pero que pueden ser bastante intrusivos. Es, por tanto, la propia compañía de publicidad la que marca los formatos de anuncios que se pueden mostrar.

Categoría	Formato	Medidas	Formato y peso máximo
Gráficos estándar	Banner	468 x 60	Gif, Jpeg, Html, Flash. 12 - 18 Kb
	Megabanner	728 x 90	Gif, Jpeg, Html, Flash. 12 - 18 Kb
	Botón	120x60, 100x100, 90x90, 80x80	Gif, Jpeg, Html, Flash. 8 - 12 Kb
	Rascacielos	160 x 600, 120 x 600, 100 x 600	Gif, Jpeg, Html, Flash. 14 - 18 Kb
	Robapáginas	300 x 250, 200 x 200. 180 x 180, 150 x 150	Gif, Jpeg, Html, Flash. 14 - 18 Kb
Gráficos especiales	Interstitial	800 x 600, 613 x 460	Gif, Jpeg, Flash. 20 - 30 Kb
	Desplegable	728 x 315, 500 x 250, 468 x 240	Flash. 100 Kb
	Layer	400 x 400	Flash. 100 Kb
	Video banner	440 x 330, 300 x 250	

Tabla 2.1: Formatos en los anuncios.

- Desplegable: Suele ser un *banner* con un formato admitido por la IAB pero que al pasar el cursor se despliega un anuncio de mayor tamaño.

- *Layer*: Este formato se caracteriza por mostrar el anuncio tapando el contenido de la página web, dejando un botón en una esquina superior para poder cerrar la pantalla.

- *Interstitial*: Este tipo de anuncio suele cargar una animación que ocupa casi toda la pantalla durante unos segundos antes de poder entrar en la página. Este tipo de publicidad es muy costosa y se puede encontrar en `Yahoo.com`, en `Marca.com` y en `Elmundo.es`.

- Vídeos de tipo *banner*: Estos vídeos tienen una ventaja sobre el resto de la publicidad y es que permiten captar mucho más la atención del usuario. Los vídeos mezclan imágenes con sonido por lo que tienen mayor impacto, éstos suelen estar enlazados con la página del anunciante. El formato a elegir irá en función del tipo de público y del tipo de campaña que se realiza, cada uno de ellos tiene su público. Por último, en la Tabla 2.1 se muestran algunos de los formatos de anuncios más frecuentes.

2.1.5. Roles en la publicidad en Internet

Como se muestra en la Figura 2.7 [26], la publicidad *online* constituye un ecosistema que incluye cuatro roles diferentes: los usuarios, los anunciantes, los editores y las redes publicitarias [31].

En el proceso de compra de un usuario mediante publicidad *online* intervienen varios roles como se muestra en la Figura 2.7. El usuario accede a la página de un editor. La red publicitaria muestra al usuario un anuncio en función de su perfil y de la página que está visitando. Cuando el usuario genera un clic en el anuncio, este le llevará a la página del anunciante que le ofrecerá su producto. Si el usuario considera que el producto se adapta a sus necesidades es probable que lo compre. En conclusión, la compañía aumenta sus ventas

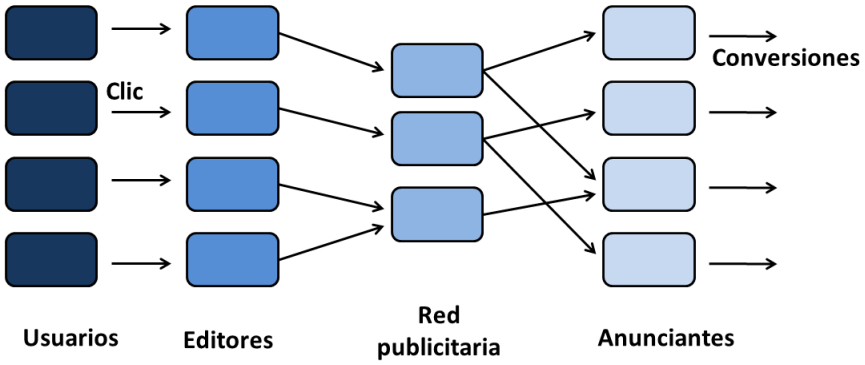

Figura 2.7: Roles que participan en la publicidad *online.*

y tanto el editor como la red publicitaria obtienen un rendimiento económico por prestar sus servicios.

2.1.6. Métodos de pago en la publicidad en Internet

Los primeros modelos de publicidad en Internet se basan principalmente en el modelo CPM, como Google Adwords en el año 2.000. El modelo CPC se impuso en poco tiempo al modelo anterior. A primera vista, es más razonable para el anunciante pagar por clics asumiendo que el usuario hace clic solo cuando le interesa.

El primero motor en pagar por palabras clave fue `GoTo.com`, que más tarde fue adquirida por Yahoo, pero sin lugar a dudas Google Adwords es actualmente el rey de este modelo, seguido de Yahoo Advertising y de Bing Ads. Los métodos de pago más extendidos en la publicidad *online* son:

- Método CPM: En los comicios de Internet el CPM fue el modelo de pago más popular en el marketing *online*. En el modelo CPM los anunciantes pagan un precio fijo por cada 1.000 impresiones de anuncio [10] tal y como se expresa en la ecuación 2.1.

$$CPM\ Valor\ del\ anuncio = CPM/1000 \tag{2.1}$$

- Método CPC: Aunque el sistema CPM se sigue usando, el modelo CPC es hoy en día el método de pago más popular [32]. El método CPC se utiliza por las principales empresas de publicidad *online*, como Google, Yahoo, Microsoft y Facebook. Los anunciantes prefieren este sistema, ya que sólo tienen que pagar cuando los usuarios están realmente interesados, es decir, cuando se genera un clic como se expresa en la

ecuación 2.2.

El fraude por clic es el principal problema en el CPC, ya que existen personas malintencionadas que tratan de confundir a las plataformas publicitarias simulando clics para obtener mayores ingresos o simplemente para causar daño al ecosistema publicitario [33]. Para evitar este problema se creó el método de pago CPA que se explica a continuación. El precio del modelo CPC se calcula como el *Click-through-rate* (CTR) multiplicado por el precio que está dispuesto a pagar el anunciante como se muestra en la ecuación 2.2:

$$CPC\ Valor\ del\ anuncio = CTR \times Precio\ Clic \tag{2.2}$$

- Método CPA: En el modelo de pago CPA los anunciantes solamente pagan cuando un usuario realiza una acción previamente acordada con la red publicitaria como el relleno de un formulario o la contratación de un servicio gracias al editor [31]. La principal ventaja de este modelo es que evita el fraude por clic, ya que los anunciantes pagan por acciones en lugar de pagar por clics. Por otro lado, el modelo de CPA tiene algunas desventajas que explican por qué su uso no está muy extendido [34].

  En este modelo existen técnicas de tipo *black-hat* que establecen como la clave del motor de búsqueda el nombre de la empresa[4]. Además, algunas campañas pretenden promover el nombre y la imagen de los productos de la compañía, lo que se conoce como *branding*. Este tipo de campañas no se centra en generar ventas de un producto, por lo que los editores difícilmente obtendrán una comisión. El valor de un anuncio se puede calcular como el valor de la comisión multiplicado por la probabilidad de que se genere una venta como se muestra en la ecuación 2.3.

$$CPA\ Valor\ del\ anuncio = Prob.Venta \times Comisión \tag{2.3}$$

- Cupones de descuento: Algunas marcas ofrecen a los editores la posibilidad de repartir cupones a través de sus páginas y cada vez que un cupón se canjea le asignan una comisión. El cupón consiste en introducir un código que identifica al editor y por el que el usuario recibe un descuento o una compensación. Este modelo de negocio existía desde hace varias décadas y exportarlo a la Red no suponía ninguna dificultad.

- Pago por tiempo: En este modelo se contacta con un editor y se acuerda mostrar un *banner* en una página durante un determinado tiempo. Las unidades más habituales

[4] Si un anunciante muestra su anuncio en un motor de búsqueda cuando un usuario escribe el nombre de la empresa es probable que el finalmente compra el producto. Obviamente esto se considera como fraude [35].

de pago suelen ser por día, por semana o por mes. Independientemente de las visitas y del rendimiento de la campaña la cantidad que percibe el editor es la misma.

- Pago por *post*: Dentro de la blogosfera, hay algunos blogs especializados en algún tema como las aplicaciones de *Iphone*, las *tablets*, o algunos blogs como `Ipadizate.es`, `AplicacionesAndroid.es` y `TecnologiaPyme.com`. Éstos pueden tener miles de *fans* y suelen tener artículos recomendando un determinado producto por lo que son muy influyentes. Algunas empresas pagan una cantidad a cambio de un artículo informando o recomendando uno de sus nuevos productos.

- Venta de enlaces: Para poder obtener un buen posicionamiento web, lo más difícil de conseguir son los enlaces. Adaptar la página web a las recomendaciones de los buscadores puede ser algo tedioso, pero no es algo difícil de conseguir. En cambio, conseguir que páginas web de calidad enlacen a una página es una tarea realmente compleja. Comprar enlaces puede ser una solución a corto/medio plazo para obtener una buena posición en los resultados de búsqueda, por lo que siempre habrá gente dispuesta a pagar por ellos. Además de esta ventaja, un enlace en una página con mucho tráfico puede traer algunos visitantes, lo cual es un valor añadido.

- Venta directa: De la misma forma que se pueden vender productos a través de un establecimiento en cualquier lugar físico, también se puede ofrecer a los clientes comprar productos a través de Internet. Existen portales de venta directa como `Redcoon.com`, que ofrecen productos de tecnología a precios competitivos, pues se ahorran gastos de alquiler, iluminación, limpieza, etc. que tendrían si tuvieran locales físicos.

  Los portales de compra pueden ser puntos de encuentro donde confluyen tanto vendedores como compradores y en ellos el portal se lleva una comisión por hacer de intermediario. El ejemplo más conocido de este tipo de negocio es eBay.

- Afiliación directa: Las marcas fuertes de Internet como Amazon, eBay, Forex, etc. tienen su propio programa de afiliación. Este programa consiste en establecer una relación directa entre el anunciante y el editor. Esto tiene como ventaja ahorrarse la comisión de la red publicitaria que hace de intermediaria, por el contrario será la marca de publicidad la que tenga que controlar el fraude y conseguir los editores de calidad.

- *Cost-per-visitor* (CPV): Este modelo se basa en que los anunciantes pagan una cantidad por cada visitante dirigido a la página web del anunciante. Es decir, al editor se le paga por cada vez que envía un usuario a la página web de un editor.

- *Pay-per-view* (PPV): También conocido como *Cost-per-view* (CPV) este modelo de publicidad se basa en las impresiones de anuncio que realmente ve el usuario. Los tipos de anuncios que suelen emplearse en este modelo son los *pop-ups*, los *pop-unders* y los anuncios de tipo *interstitial*.

  Los anuncios que no fueron visualizados, aunque se hayan cargado en la página, no se consideran como parte del pago por visión. Estos podrían ser los que están a pie de página, pero que el usuario al no haberse desplazado hasta la parte inferior de la página, no ha logrado visualizar.

- *Cost-Per-Click-Play-View* (CPCPV) y *Cost-Per-Completed-View* (CPCV): Son dos modelos en los que los anunciantes pagan cuando un usuario empieza a visualizar el vídeo. En el primer caso, cuando el usuario ve más del 80 % del vídeo o el vídeo completo en el segundo.

2.1.7. Las redes CPM, CPC y CPA en la publicidad en Internet

Los tres principales modelos de publicidad en internet son las redes CPM, CPC y CPA. El modelo CPM tuvo una gran importancia en los inicios de la publicidad en internet. El modelo CPC, también llamado como *Pay-Per-Click* (PPC), es un modelo de publicidad en el que el anunciante paga una cantidad concreta al editor cada vez que un internauta hace clic sobre el anuncio [36]. Y el modelo CPA a pesar de presentarse como novedoso y sin el eterno problema del fraude por clic, no consigue un gran nivel de aceptación. En las siguientes líneas se describe cada una de las principales redes con mayor profundidad:

2.1.7.1. Las redes CPM

En los primeros días de Internet el primer modelo de pago adoptado en el marketing *online* fue *Cost-per-mille* (CPM). Es uno de los modelos más sencillos y consiste en pagar al editor por cada mil impresiones de anuncio. Un CPM de un euro significa que el anunciante paga un euro por cada 1.000 veces que sus anuncios aparecen en la página web [10], tal como se expresa en la ecuación 2.4. Y el costo de la campaña es la suma del precio de todas impresiones de anuncio, como se define en la ecuación 2.5.

$$Valor\ Anuncio\, CPM = Costo\, CPM/1{,}000 \tag{2.4}$$

$$Costo\ campaña\, CPM = \sum_{i=1}^{N} CPM\ Valor\ anuncio\,(i) \tag{2.5}$$

Sin embargo, algunas de las impresiones pueden no ser contabilizadas. Esto sucede cuando accede el propio editor a su página o cuando hay una recarga interna. Hay personas que tienen pocas visitas y que no saben bien dónde colocar los anuncios dentro de la página para generar clics. Éstas preferirán este sistema porque en CPC o en CPA se obtienen muy pocos beneficios. Dentro del *marketing* digital, hay un concepto que se conoce como *branding* que consiste en aumentar el valor de la marca. Cuando se hace una campaña de *branding* se trata de añadir valor a la marca que se anuncia, no importan tanto el corto plazo, es decir, las ventas que pueden generarse en esa campaña, sino conseguir que las personas conozcan la marca y tengan una buena imagen de ella en el futuro.

2.1.7.2. Las redes CPC

Este modelo es el más extendido, principalmente porque las grandes marcas de publicidad se han decantado por este sistema. El precio del clic oscila entre los 0,05 dólares y 1 dólar, aunque algunas palabras clave pueden alcanzar valores superiores a los 50 dólares.

El precio por clic está en función de la demanda, es decir, del número de personas que pujan por esa palabra clave. Cuanto mayores sean los ingresos que se puedan obtener por el anuncio mayor será el precio que estarán dispuestos a pagar los anunciantes. Es distinto el precio de un anuncio de un seguro de coche que el de un anuncio de comida para pájaros. Lógicamente, el segundo anuncio será mucho más barato, pues el margen de beneficio del producto es mucho menor. En el primer caso, el clic puede tener un coste de 1,20$ y en el segundo de 0,06$.

Este modelo está más orientado a conseguir conversiones. Para conseguir vender un producto existen tres factores clave: hacer una buena campaña, tener una página web optimizada y ofrecer un producto competitivo en calidad y en precio. Los expertos recomiendan tener una página de aterrizaje[5] (*landing page*) orientada a que el usuario haga una conversión. Es interesante documentarse sobre algunas estadísticas de estas páginas [37].

En el modelo CPC los anuncios se muestran en función de varios parámetros como son las características personales del usuario que navega, las palabras clave que éste ha buscado o el contenido de la página en que se muestra el anuncio. La calidad de un anuncio para algunas redes CPC consiste en el famoso CTR, que es el porcentaje de veces que se hace clic sobre el anuncio multiplicado por el precio del clic, que es lo mismo que la rentabilidad económica que le genera a la red publicitaria y se calcula con la ecuación 2.6.

$$Ranking\ Anuncio = CPC\ \times\ Puntuación\ Calidad \tag{2.6}$$

[5]La *landing page* es la página donde aterrizan los usuarios tras hacer clic en el enlace de publicidad. Es muy importante optimizarla ya que de ello depende en gran medida que el usuario termine comprando [37].

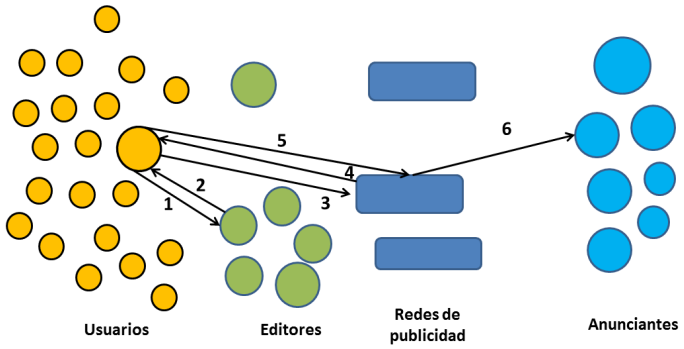

Figura 2.8: Pasos para mostrar un anuncio en el modelo CPC.

Muchas veces los servidores de anuncios ralentizan la velocidad de carga de una página, consumen demasiado ancho de banda y extraen información confidencial de los usuarios, lo cual perjudica al ecosistema publicitario.

Aunque este sistema se sigue utilizando, el método CPC es actualmente el método de pago más popular [32] y es utilizado por empresas de publicidad *online* como Google, Yahoo, Microsoft y Facebook. Estos anunciantes prefieren el método de pago CPC ya que pagan solamente cuando los usuarios están realmente interesados en sus productos, es decir, cuando realmente hacen clic en ellos.

El precio de un clic en algunas plataformas CPC es fijo. En otras plataformas CPC, el anunciante indica un precio máximo y el precio del clic varía según determinados parámetros como la calidad del editor, el máximo CPC establecido por otros anunciantes o el comportamiento de los usuarios cuando visitan un sitio web.

El precio de un anuncio puede expresarse mediante la ecuación 2.7. Como el anunciante paga únicamente cuando se genera un clic, se multiplica el precio por clic por el CTR y el costo de la campaña será la suma del costo de todos los clics, tal y como se expresa en la ecuación 2.8. Sin embargo, el principal problema del modelo CPC es el fraude por clic[6].

$$Valor\ Anuncio\ CPC = CTR \times Precio\ clic \tag{2.7}$$

$$Costo\ campaña\ CPC = \sum_{i=1}^{N} Precio\ clic\ (i) \tag{2.8}$$

En la Figura 2.8 se pueden ver los pasos en el modelo publicitario CPC:

[6]Hay personas que tratando de engañar a plataformas publicitarias que simula clics para conseguir más ingresos o simplemente para causar daños a las plataformas de publicidad *online* [33].

1. Un usuario escribe en la barra del navegador la URL de la página.

2. La página que se descarga tiene tanto el contenido de la página como el código proporcionado por la red publicitaria. Lo cual tiene la ventaja de que aunque el anunciante no tenga un gran ancho de banda no afecta al usuario porque es la red publicitaria la que sirve al anuncio.

3. Normalmente el navegador carga un *javascript* en el servidor. Este código al ejecutarse recoge algunos parámetros del usuario como la IP, las *cookies*, la hora, la versión, el navegador, etc. Esta forma permite ejecutar un código complejo invocando al servidor del anuncio con pocas líneas de código.

4. Toda esta información es recibida por el servidor de la plataforma de publicidad con su consiguiente petición de anuncios.

5. El servidor de anuncios ejecutará un algoritmo que permitirá asociar el mejor anuncio en función de estos parámetros en un tiempo corto.

2.1.7.3. Las redes CPA

En el modelo CPA los anunciantes sólo pagan cuando un usuario realiza una acción como rellenar un formulario o contratar un servicio [31]. La principal ventaja de este modelo radica en su capacidad para evitar el fraude por clic, pues los anunciantes pagan sólo por acciones llevadas a cabo por un usuario en vez de sólo por clics.

En el modelo CPA un editor recibe una comisión por cada acción realizada por un usuario asociado a su identificador. Los editores se registran en la red CPA y solicitan participar en las campañas, el anunciante puede aceptar o rechazar a los editores. Algunas redes valoran a los anunciantes para que los editores sepan qué campañas tienen más éxito y a los editores para que los anunciantes sepan sobre su reputación.

Sin embargo, el modelo CPA también adolece de algunas desventajas que explican por qué su uso no está muy extendido. Las técnicas *black-hat* o fraudulentas también se encuentran en este modelo[7] como *cookie stuffing*[8] y palabras clave del motor de búsqueda prohibidas [35]. Además, algunas campañas simplemente pretenden promover el nombre y la imagen de la empresa, conocida como *branding*. Ya que estas campañas no están orientadas a la venta de un producto [39].

[7]Las técnicas fraudulentas consisten principalmente a hacer trampa para ganar más dinero. Éstos se pueden también utilizar para dañar la campaña de un anunciante para el ecosistema de la publicidad [38].

[8]Esta técnica consiste en dejar una cookie en la memoria caché de los equipos de los usuarios que ven un anuncio. Hacer creer al anunciante que hicieron un clic en este anuncio. Así, si el usuario compra el producto anunciado el timador se llevará una comisión [38].

El valor de un anuncio en el modelo de CPA puede estimarse como la multiplicación del CTR, el valor de la comisión y la probabilidad de que se genere una venta, tal y como se expresa en la ecuación 2.9. El costo de la campaña puede expresarse como la suma de todas las comisiones, como se define en 2.10.

$$Valor\ Anuncio\ CPA = CTR \times Prob.\ de\ venta \times Precio\ Comisión \tag{2.9}$$

$$Costo\ campaña\ CPA = \sum_{i=1}^{N} Precio\ Comisión\ (i) \tag{2.10}$$

El anunciante permite a los editores promocionar un producto a través de *banners*, enlaces en páginas web, enlaces patrocinados en buscadores (SEM), códigos promocionales en sitios especializados, correos electrónicos, aplicaciones de software o redes sociales. Para poder controlar el uso de acciones se emplean *cookies*[9], que son archivos que se depositan en la memoria de los navegadores y que identifican de manera unívoca al editor.

Este sistema tiene algunas limitaciones como que un usuario haga una compra desde un navegador diferente al que visitó la página la primera vez y por lo tanto, no se pueda asociar las *cookies* con el usuario. Si el usuario simplemente visualizó el anuncio, se le dejará una *cookie post-view*, en cambio si el usuario hizo clic en el anuncio el usuario tendrá una *cookie post-click*.

Las *cookies* tienen algunas limitaciones, una de ellas es que si el usuario navega de manera anónima no se le guardan las *cookies*. Otra es que si el usuario borra la caché, también se borrarán todas las *cookies*. Y si utiliza un navegador para ver la oferta, pero la compra se hace desde otro ordenador, esta venta no quedará registrada. Por lo tanto, siempre hay un porcentaje de ventas/registros que se pierden. En la Figura 2.9 se ven los pasos que se siguen para poder mostrar un anuncio.

A continuación se explica en detalle cada uno de los pasos:

1. Un usuario accede a una página web donde hay un *banner* de una red CPA, ya sea directamente o a través de un buscador.

2. La página que se descarga tiene tanto el contenido HTML de la página web como el código *javascript* que invoca a la red CPA para que sirva el anuncio.

[9]Las *cookies* son "cadenas de información alfanumérica formadas por diferentes campos específicos que son depositadas por el servidor en el disco duro del cliente durante una visita del mismo". Las *cookies* son pequeños archivos de texto que son enviadas desde el servidor al navegador del cliente y que se almacenan en la memoria de la computadora. Tiene diversos usos como sesión, información sobre la fecha, etc.

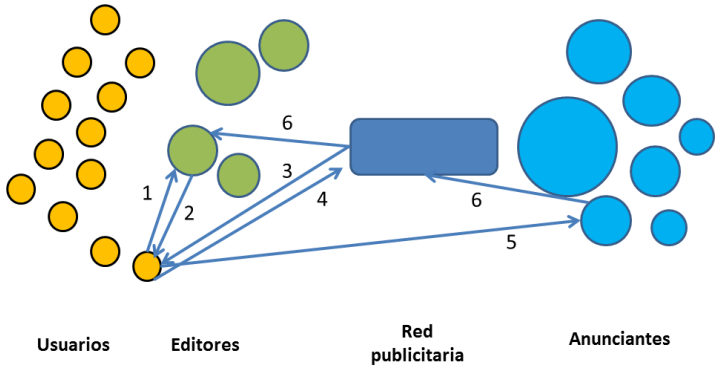

Figura 2.9: Pasos de una comisión en las redes CPA.

3. La red CPA mostrará el *banner* y dejará una *cookie post-view* en la caché del usuario, indicando que ese usuario ha visualizado el anuncio. En caso de que el usuario se decida hacer clic en el *banner* además de la *cookie post-view* se le dejará una *cookie post-click*.

4. El código que se ejecuta puede analizar algunos parámetros del usuario como son la dirección IP, el historial de búsqueda, la fecha de acceso, el tipo de navegador, etc. Esto permite recoger ciertos parámetros que se pueden usar para detectar un posible fraude o para ver el perfil más frecuente de un usuario.

5. Si el usuario hace una compra, en caso de tener una *cookie post-view* se le asignará una comisión. Si tuviera una *cookie post-click*, se le asignará una comisión superior.

6. El anunciante aprueba la venta y paga la comisión a la red CPA que le otorgará al editor su parte correspondiente.

2.1.8. Fraude en la publicidad en Internet

La publicidad *online* se ha convertido en una industria poderosa que constituye la principal fuente de ingresos de las empresas más importantes de Internet. El enorme volumen de facturación de esta industria supone un reclamo atractivo para todo tipo de ciberdelincuentes, por lo que los fraudes en torno a la publicidad en Internet se han multiplicado en los últimos años [40].

Dependiendo del tipo de modelo publicitario se puede engañar de distintas formas. En el modelo CPM, el fraude consistirá en multiplicar artificialmente el número de impresiones[41]; en el modelo CPC, el engaño consistirá en aumentar artificialmente el número de clics

[36, 42] y en el modelo CPA el objetivo será reproducir las acciones deseadas por el anunciante como descargas de software o rellenos de formularios sin que acciones estén en absoluto vinculadas con el anuncio en cuestión[31, 43].

Los fraudes pueden ser cometidos por personas de modo libre y desinteresado o a cambio de una contraprestación económica, o por *botnets*[10] [33]. Los principales responsables de los fraudes suelen ser los editores, que mediante diferentes técnicas fraudulentas consiguen obtener más ingresos simulando tráfico, clics o acciones. También pueden ser los competidores de los anunciantes para perjudicar a las empresas rivales. Finalmente, las culpables pueden ser las plataformas publicitarias que no persiguen los fraudes debido a que se benefician económicamente de estos. Los fraudes multiplican los datos de la publicidad *online*, y con ellos los beneficios de la propia plataforma intermediaria.

Las principales razones por las que el fraude está tan extendido son:

- Carencia de una legislación adecuada: El fraude se puede realizar desde países como la India o Nigeria donde no existe una legislación al respecto, por lo que las repercusiones legales no suelen ser un freno para los editores. En la mayoría de los casos, lo único que puede suceder es que el editor sea expulsado de la plataforma. Por otro lado, un proceso judicial sería algo lento y complejo, y algo que las redes preferirán evitar pues pondría en evidencia que han sido vulnerados.

- Conflicto de intereses: En casi todas las redes hay un conflicto de intereses, ya que el que detecta el fraude y el que se beneficia indirectamente si éste no es detectado, es la misma red publicitaria. Por lo tanto, el peligro de las redes publicitarias reside tanto en que los editores cometan fraude como en que la plataforma de publicidad no ponga los medios para detectarlo.

- Desventaja competitiva: Una empresa que permita hacer trampas, especialmente si es del modelo CPA, tendrá un volumen de facturación mayor. A muchas compañías no les interesa trabajar con redes que generen pocos registros o ventas. Si se permite el fraude, muchos de los usuarios que lo cometen y que son expulsados de otras plataformas irán a parar a la nueva plataforma, permitiendo así aumentar los beneficios.

- Comunicación entre las plataformas: Si una plataforma sospecha ser víctima de un engaño dejará de participar en el programa, pero rara vez informará a las otras. No es una buena publicidad para una empresa haber sido víctima de un fraude por lo que no suelen denunciar [34].

[10] Los *botnets* son robots que simulan la conducta de un internauta y repiten acciones previamente programadas [44, 45].

- Caducidad de cuentas: Las cuentas que deja un editor se borran a los pocos años, por lo tanto, no queda rastro de si las redes cometieron fraude en el pasado.

2.1.8.1. Fraudes en el modelo CPM

Las principales trampas en los modelos CPM están dirigidas a aumentar artificialmente el número de impresiones de una determinada página de manera que el anuncio sea mostrado en más ocasiones y de que el anunciante tenga que pagar una cantidad superior al editor de la página web.

Se trata, por tanto, de falsear datos, haciendo pagar al anunciante por anuncios que han sido desplegados ante un robot o un usuario sin la suficiente calidad, bien porque ha sido mostrado durante un intervalo de tiempo muy breve o bien porque no ha podido ser visualizado con la suficiente claridad.

Dos son las formas de incrementar el tráfico: programando *botnets* que visiten muchas veces determinadas páginas, o bien infectando con un software malicioso miles de ordenadores de manera que visiten sin consentimiento del usuario páginas de una lista, que se va actualizando conforme a la conveniencia del estafador.

Entre los métodos más destacados de fraude en las redes CPM o redes *Pay-per-view* (PPV) se encuentran: contratar a empleados para que vean páginas y hagan clic en los anuncios, contratar a terceras partes para que hagan esto mismo, hacer esquemas piramidales con este objetivo o usar *botnets* para que hagan clics e impresiones.

2.1.8.2. Fraudes en el modelo CPC

En los modelos CPC las trampas consisten en la producción de los llamados clics fraudulentos: estos clics no provienen de potenciales consumidores, sino de agentes ilegítimos con la finalidad de engañar al sistema para obtener un beneficio económico a costa de un anunciante.

Las formas de aparición y las motivaciones de estos clics fraudulentos son muy variadas, pero básicamente pueden provenir de tres sujetos: el editor, la competencia del anunciante, o personas que quieren perjudicar todo el ecosistema publicitario *online*.

Para saber si el clic es válido habría que conocer la intención del usuario, lo cual es muy difícil. Cuando una persona se conecta a una web, encuentra un *banner* de su interés y se decide a hacer clic. Una vez que la página del anunciante, observa el producto detenidamente y se acuerda de que tiene algo pendiente pasando a otra tarea. Seguir este mismo patrón de comportamiento para engañar a la plataforma de publicidad es algo sencillo y difícilmente diferenciable del primer caso.

Primeramente, se debe distinguir entre un clic inválido y uno fraudulento. Un clic inválido ocurre cuando no existe mala intención, como cuando un usuario tiene el hábito de hacer doble clic en un anuncio, en este caso el usuario no tiene intención de engañar.

Un clic fraudulento es el que se comete cuando un editor hace clic en sus propios anuncios para aumentar los ingresos. Hacer clics falsos es una tentación que tienen muchos editores. Muchas personas quieren aumentar sus ingresos y se desesperan al comprobar que éstos no lo hacen. Ante estas situaciones es muy frecuente que se generen clics artificiales. Si una plataforma no es capaz de detectar estos ataques, no será interesante para los anunciantes.

A continuación se describen las principales trampas en el modelo de publicidad CPC:

- Inflación de clics: Se trata de un editor que, personalmente o a través de amigos, hace clic en los anuncios de su página para aumentar sus ingresos. Estos clics son fácilmente detectables, su relevancia económica es pequeña, y las plataformas publicitarias no suelen emprender acciones legales, más allá de no cobrar al anunciante y expulsar al infractor de la plataforma.

- Clics a la competencia: En este caso un competidor de un anunciante provoca clics en la campaña de éste hasta agotar su presupuesto publicitario y poder situar sus propios anuncios en una posición preferente o simplemente para evitar que gane dinero. Difícilmente se puede hacer algo contra estas personas, ya que no es fácil saber su identidad y tampoco son cantidades que justifiquen tomar acciones legales.

- Clics contra un editor: Se trata de perjudicar a un editor haciendo clics indiscriminados en sus enlaces con la intención de que la plataforma publicitaria crea que se trata de un editor tramposo y sea expulsado.

- Clics solicitados por el editor: Algunos editores buscan la complicidad de los internautas, y les solicitan de un modo más o menos indirecto que contribuyan con su web haciendo clic en la publicidad empleando mensajes del estilo: “Si te gusta la web, haz clic en la publicidad”. El internauta no está interesado en la publicidad, con lo que no es un cliente potencial del anunciante.

- Clics forzosos: Para aumentar el número de clics, el editor puede recurrir a técnicas que obstaculicen la libertad de navegación del usuario, y le lleven a hacer clic en anuncios que no le interesan. Esto puede llevarse a cabo, en primer lugar, camuflando o disfrazando los anuncios, de modo que parezcan parte del menú de la web, superponiendo un *banner* a una determinada información.

  También puede modificarse el código para que el anuncio se abra al hacer clic en cualquier enlace de la página. Otra posibilidad es situar los enlaces de los anuncios

muy próximos a lugares donde el usuario tenga que hacer clic, para facilitar su error y el consiguiente clic en el anuncio. Esto ocurre por ejemplo, en un juego en el que haya que hacer clic en diversas partes de la pantalla.

- Granjas de clics: Las granjas de clics (*click-farms*) son grupos de personas que están capacitados para generar clics en los anuncios en una campaña determinada, con el fin de dañar el propietario [46]. Normalmente están situados en países donde las personas perciben salarios muy bajos. Hace ya años una investigación publicada en *"The Times of India"* alertaba sobre este tipo de fraudes publicitarios [47].

- *Botnets* generadores de clics: Estos robots simulan el comportamiento de una persona en Internet y generan clics en anuncios de forma automática. Uno de los *botnets* más perjudiciales descubiertos hasta la fecha ha sido el *Botnet Chameleon*. Se estima que infectó más de 120.000 ordenadores de todo el mundo, simulando 9.000 millones de clics fraudulentos al mes, y produciendo un perjuicio estimado de 6 millones de dólares mensuales a las agencias publicitarias y a los anunciantes [44].

  Los *click-bots* son programas malintencionados que normalmente se instalan en el ordenador del usuario y que generan un clic automáticamente para perjudicar ecosistema publicitario [48]. Los *click-bots* a la carta son un software especializado en hacer trampas que se puede comprar. Se les puede dar una lista de páginas sobre las que hacer clics y de *proxies* para que vayan cambiando la dirección IP.

- Enemigos de la plataforma: Si una plataforma está consiguiendo muchos anunciantes, una forma de hacer que ésta fracase es haciendo clics falsos y así los anunciantes no estarán satisfechos. Algunos interesados en que esta plataforma fracase pueden ser su competencia.

- Errores de la plataforma: Es posible que una plataforma tenga un error al colaborar con terceros y que cuente varias veces un mismo clic.

- Usuarios que hacen doble clic: Algunos usuarios siguen con la costumbre de hacer doble clic para abrir un enlace. Aunque se supone que con el paso del tiempo estos cada vez serán menos.

- *Scripts* automáticos: Hay editores que contratan tráfico artificial de baja calidad pero que genera clics. Este tráfico se genera con programas que hacen clic automáticamente en los anuncios de pago. De esto se benefician los competidores de la plataforma, ya que sus clientes se sienten engañados por el bajo rendimiento.

- Vandalismo: Algunas personas hacen clic en los anuncios de publicidad de ciertas empresas simplemente porque les parece divertido causar un perjuicio a un tercero. Si estas personas tienen conocimientos avanzados podrán desarrollar *click-bots* o *scripts* para hacer clic de forma automática.

- Falsear palabras clave: No todos los anuncios tienen el mismo precio, en gran parte depende de la temática. Esta técnica consiste en escribir palabras ocultas mediante etiquetas HTML que sean invisibles a los usuarios pero no a los buscadores. Si el fondo es blanco, se pueden poner letras en blanco que digan: "Alquilar coche", "Coches de lujo", "Comprar coches gama alta", etc. de forma que aparezcan anuncios con un alto CPC y aumenten los ingresos.

2.1.8.3. Fraudes en el modelo CPA

Algunos editores pueden realizar pequeñas trampas para poner a prueba a la plataforma. Normalmente, en caso de no ser detectados, aumentan paulatinamente el nivel de fraude para intentar alcanzar mayores ganancias. Entre las trampas más conocidas en el modelo CPA se podría destacar:

- Relleno automático de formularios: Esta técnica trata de conseguir registros mediante usos no lícitos, como *botnets* que descarguen programas o que rellenen formularios de forma automática. También pueden ser los propios afiliados, utilizando *proxies* para evitar ser detectados o puede realizarse a través de terceros por otros medios como contratando empresas, premiando a los usuarios por realizar registros o extorsionándolos mediante unas fotos comprometidas.

- *Malware*: Algunos programas se instalan en ordenadores sin consentimiento del usuario. Posteriormente abren ventanas con páginas de redes CPA. Estos programas suelen expandirse mediante *malvertisement*.

- SEM ilegal: Si en una campaña se permite utilizar marketing en buscadores (*Search engine marketing*) (SEM), el editor podría contratar como palabra clave el nombre de la marca, lo que dará unos resultados excesivamente buenos. Ya que cuando alguien quiere comprar un producto suele escribir el nombre de la marca en el buscador. Como es lógico, el uso de estas palabras clave en SEM está terminantemente prohibido por el anunciante, pero aun así, algunos editores lo hacen.

- *Cookie stuffing*: Mediante esta técnica, un editor deja una *cookie* en el caché de los navegadores de los usuarios sin su consentimiento, simulando que han visitado su

página. Para poder obtener unos ingresos altos es necesario aplicarlo en cientos de miles de usuarios.

- *Pop-up* y *pop-under*: El *pop-up* consiste en modificar el código del anuncio para abrir una página de destino de forma que quede una *cookie* en la caché del navegador del usuario y se produzca un efecto similar al *cookie stuffing*. Los *pop-under* abren una nueva ventana que se coloca detrás sin que el usuario lo perciba. Solamente al cerrar la ventana del navegador principal, quedaría esta segunda ventana y no se sabe en qué momento de la navegación apareció.

- Bases de datos ilegales: Algunos editores utilizan bases de datos compradas de forma ilegal. Suelen ser bases de datos que se compraron de manera legal pero que se comercializan por terceros a un menor precio sin el consentimiento del autor. Los editores emplean servidores SMTP para mandar emails a cientos de miles de usuarios esperando que alguno rellene el formulario.

- Impresiones falsas: Para conseguir la comisión *post-view* o para disimular los ratios obtenidos al aplicar algunas técnicas *black-hat* se hacen varias impresiones cuando se muestra un anuncio, dejando varias *cookies* en lugar de una.

2.1.9. Cálculo del valor de un anuncio en la publicidad en internet

Los dos modelos de pago más extendidos son el CPM y el CPC. En el modelo CPM los anunciantes pagan cierta cantidad cada vez que un usuario accede a una página web y su anuncio se muestra. En los anunciantes del modelo CPC sólo se paga cuando un usuario hace clic en un anuncio [49].

El costo de los anuncios y la estimación de la rentabilidad son factores determinantes en el mercado *online*. Predecir la rentabilidad de un anuncio permite que los anunciantes, los editores y las redes publicitarias obtengan el máximo beneficio. En esta área de investigación, los algoritmos, las metodologías y las técnicas están en constante evolución.

En los últimos años han surgido nuevos enfoques en el marketing digital como modelos de decisión o como sistemas basados en la rentabilidad del visitante [50]. Otros estudios afirman que los anuncios CPC tienen un efecto directo en la captación de clientes y en la efectividad de los anuncios *online* [51]. Puesto que la rentabilidad está muy relacionada con que el usuario haga clic, estimar correctamente el CTR de un anuncio en un sistema es un factor importante.

El valor CTR se puede calcular de varias maneras, tanto para los anuncios que aparecen con frecuencia como para los que se muestran por primera vez. Algunos modelos aplican

una regresión logística para predecir el CTR de los anuncios. Las regresiones logísticas tienen una gran capacidad para representar y construir condiciones con facilidad [32, 52].

Otros enfoques en este tema aplican algoritmos de predicción de CTR basado en diferentes técnicas de regresión lineal multicriterio (*Multi-Criteria Linear Regression*) (MCLR), regresión con múltiples criterios basados en kernel (*Kernel-based Multiple Criteria Regression*) (KMCR), programación lineal multicriterio con regresión (*Multi-criteria Linear Programming*) (MCLP) y basados en múltiples criterios de programación basados en kernel (*Kernel-based Multiple Criteria Programming*) (KMCP)[53].

Otros sistemas comerciales como el motor de búsqueda Bing de Microsoft utiliza un algoritmo para la predicción del CTR basado en redes bayesianas [54]. Los sistemas de recomendación también desempeñan un papel importante en la maximización de los beneficios. La selección del mejor anuncio para que se muestre en un espacio es uno de los principales problemas de la publicidad *online*. La mayor parte de las técnicas recomendadas para estimar el CTR de un anuncio se basa en los patrones que determinan el comportamiento de los usuarios.

Algunos recomiendan las redes neuronales que son modelos que aprenden de datos con pocas dimensiones y se pueden mejorar consultando listas con preguntas relacionadas con búsquedas del vecino más cercano [55]. Otras recomendaciones proponen modelos basados en espacios temporales para estimar el CTR en el contexto de recomendación de contenidos [56].

La detección de *spam* y *malware* es también una cuestión importante para las actividades de publicidad y no sólo en los canales de publicidad sino también en los nuevos canales de las empresas de comercio electrónico con nuevos *gadgets* como *smartphones* y tabletas que se han introducido en el mercado [57]. Se han realizado numerosos esfuerzos para detectar *spam* y también se han aplicado algoritmos de clasificación de minería de datos [58, 59].

En cuanto a la predicción de las ventas, se han producido numerosos avances. Uno de los métodos más utilizados en el pasado, es el *Auto Regressive Integrated Moving Average* (ARIMA). Otros métodos estadísticos utilizados para predecir ventas analizan la probabilidad de un suceso y el número de veces que se repetirá. La teoría de juegos se basa en el estudio del comportamiento de las empresas con respecto a otras en un determinado nicho de mercado [60].

Sin embargo, estos métodos han sido ya superados por los métodos supervisados de ML que tienen en cuenta factores como la estacionalidad y el clima, que se refieren a las ventas en el futuro. Estos métodos pueden establecer una relación entre el conjunto de entradas y el de salidas. Algunos de ellos son interpretables, es decir, pueden indicar qué entrada ha tenido una mayor influencia en los resultados predichos. Algunos métodos supervisados

son los *splines* cúbicos, los modelos basados en árboles, las redes bayesianas, los modelos *Random Forest* (RF) y las redes neuronales (*Neural Networks*) (NN) [61].

2.2. Minería de datos

Los avances informáticos permiten almacenar grandes cantidades de datos. Basta imaginar la cantidad de datos que se puede almacenar sobre los 3 mil millones de usuarios que navegan por la Red. Sobre estos usuarios se puede almacenar el navegador que utilizan, las palabras clave que los usuarios introducen en los buscadores, las páginas que visitan y el tiempo que están en las páginas y muchos más datos [62].

También pueden registrarse datos mucho más personales como las horas a las que se conectan, las IPs que utilizan, la localización geográfica de los dispositivos, los gustos que tienen en Facebook, las fotos que suben o los correos que mandan y a qué personas. Muchos de estos datos están cifrados y no pueden registrarse porque violan el derecho de intimidad pero muchos otros si se pueden registrar. Como es lógico, se requiere de varios terabytes diarios para almacenar tanta información. Pero todos estos datos y esta información no serviría de nada si no se pudiera procesar; por lo tanto, para poder analizar y sacar conclusiones de un volumen de información tan grande se requiere la minería de datos (*Data Mining*) (DM) [63].

La aplicación de métodos de aprendizaje automático, a grandes bases de datos se llama minería de datos[11]. La minería de datos es un campo de la estadística y las ciencias de la computación referido al "proceso que intenta descubrir patrones en grandes volúmenes de conjuntos de datos" [64]. Estos patrones permiten hacer predicciones asumiendo que el futuro próximo no será muy distinto al pasado. A veces, la cantidad de información es tan grande que solamente puede ser procesada por una gran computadora para extraer conclusiones en un tiempo razonable. La MD se aplica cada vez a más campos como es la predicción de ventas, la detección de fraude en tarjetas bancarias, la estimación de probabilidades de venta de un producto, la selección del anuncio mostrar para un determinado usuario, las recomendaciones de productos a un usuario en base a su perfil, las predicciones meteorológicas y un largo etcétera [65].

La minería de datos es un campo interdisciplinario de las ciencias de la computación que consiste en el proceso de descubrir patrones en conjuntos de datos mediante métodos de inteligencia artificial, análisis matemático, estadística y sistemas de bases de datos. La minería de datos es el análisis del proceso de conocimiento extraído de las bases de datos

[11] El nombre de minería de datos viene de una analogía a la minería donde se extraen toneladas de materiales para procesarlos y quedarse con una pequeña cantidad de material muy valioso.

(*Knowledge Discovery in Databases*) (*KDD*).

El objetivo general de la minería de datos consiste en extraer información de una base de datos y convertirla en una estructura comprensible para su uso posterior. La minería de datos tiene seis fases principales: detección de anomalías, reglas de asociación para el aprendizaje, *clustering*, clasificación, regresión y simplificación.

2.2.1. Aprendizaje automático

Arthur Samuel definió el aprendizaje automático en 1959 como: "El campo de estudio que permite a las computadoras aprender sin ser explícitamente programadas" [66]. ML es una rama de la inteligencia artificial que utiliza muchas disciplinas de las matemáticas tales como estadística, reconocimiento de patrones, lógica o modelos evolutivos.

El aprendizaje automático es una de las fases de la minería de datos y es una parte importante de la Inteligencia Artificial. La Inteligencia Artificial dota a los sistemas de la capacidad de aprender. Es decir, de hacer capaz al sistema de adaptarse a cambios, de predecir el comportamiento futuro y de tomar decisiones acertadas de forma independiente.

Para que una máquina “aprenda”, se suelen utilizar una base de datos que consiste en un conjunto de registros con datos que representan las entradas del modelo. Por ejemplo, se puede registrar de un usuario: su altura, su género, su edad, su nivel adquisitivo, etc. Estos datos representan los atributos del sujeto y los se utilizan como entradas del modelo. La salida del modelo sería si el usuario compró o no compró un determinado producto.

Normalmente se utiliza un conjunto de muestras para entrenar el modelo que se conoce como datos de entrenamiento (*training set*). Y se reserva un conjunto de muestras llamado datos de prueba (*testing set*). Los datos de entrenamiento se utilizan para construir el modelo y los de prueba para evaluarlo.

Para evaluar el modelo se utilizan distintas métricas como la precisión, el RMSE, la sensibilidad, la especificidad, etc. que se explican más adelante [67]. Los modelos pueden ser predictivos, cuando predicen el comportamiento futuro, o descriptivos, cuando tratan de crear reglas de comportamiento a partir de los datos.

El aprendizaje automático también se utiliza para formar módulos que se utilizan en la toma de decisiones. Muchas veces se tienen millones de datos pero es muy difícil extraer reglas o encontrar patrones a partir de toda esta información. El aprendizaje automático puede ayudarnos a resumir la realidad y a extraer reglas sencillas que ayuden a crear patrones de comportamiento.

Un ejemplo de regla sería: Si el cliente es soltero, es su primer trabajo y es extranjero la probabilidad de que pague el crédito es de un 0,42 %. Teniendo esto en cuenta, los créditos para estas personas estarán más restringidos o tendrán unos intereses más altos. Esto también

tiene gran utilidad en la publicidad *online* ya que permite dirigir estas campañas a aquellos usuarios que con mayor probabilidad comprarán nuestros productos.

El aprendizaje automático puede servir para detectar anomalías [68]. Para ello, se entrena el modelo con el comportamiento de los usuarios a través de la página. Cuando el sistema detecta que muchos usuarios tienen patrones diferentes a lo "normal" puede mandar un mensaje de alarma pues puede haber un intento de acceso por parte de un *hacker*. El aprendizaje automático ha sido aplicado con éxito en muchos problemas de reconocimiento de voz y de imágenes y en la robótica [69].

El aprendizaje automático utiliza muchas disciplinas de las matemáticas como estadística, reconocimiento de patrones, lógica, inteligencia artificial y modelos evolutivos. Lo que se busca son algoritmos que generen modelos lo más precisos posibles y que a la vez requieran de un tiempo de ejecución y de un tamaño de memoria lo más reducido posible. Además se puede mejorar con otras características como que sean resultados interpretables, que tengan resistencia al ruido y que sean estables [70].

Los modelos tradicionales han sido reemplazados por técnicas de ML como redes neuronales artificiales (Artificial Neural Networks) (ANN) [71]. Estas redes están compuestas de nodos interconectados organizados en varias capas. Cada nodo tiene un peso que se ajusta mediante un algoritmo de entrenamiento que puede utilizar *"Back-propagation"*. Esto permite a las ANN predecir ventas futuras de forma mucho más precisa.

Varias investigaciones destacan la importancia de las técnicas de ML para una empresa y para el desarrollo empresarial [72, 73]. Una predicción correcta del número de coches que se venderán puede ser crucial en la industria automotriz [73]. Los métodos multivariable de regresión lineal (MLR) y la *Support Vector Machine* (SVM) han sido ampliamente utilizados, así como *Decision Trees* (DT), *k*-NN y RF [74]. Muchos investigadores recomiendan las ANN [75], específicamente el perceptrón como el mejor método [76].

2.2.2. El aprendizaje supervisado

En el aprendizaje supervisado los problemas tienen una entrada X y una salida Y. Estos modelos tratan de crear una relación entre las entradas y las salidas. Estos problemas son de clasificación cuando la salida es una clase de un grupo, y de regresión cuando se trata de predecir un valor real que puede ser continuo [77].

En el aprendizaje automático existe un modelo definido por un conjunto de parámetros: $y = g(x|\theta)$, donde $g(\cdot)$ es el modelo y θ son sus parámetros. En los métodos de regresión Y es un número real, y en los métodos de clasificación Y es una clase. La función discriminante $g(\cdot)$ separa las instancias de las diferentes clases. El programa de aprendizaje automático optimiza los parámetros reduciendo al mínimo el error de aproximación, es decir, las esti-

maciones deben ajustarse lo más posible a los valores del conjunto de entrenamiento.

2.2.2.1. Métodos supervisados de clasificación

El objetivo de la clasificación es aprender una asociación entre las entradas X y las salidas Y, donde $Y \in \{1, ..., C\}$, donde C es el número de clases. Cuando solamente hay dos posibles salidas, es decir, $C = 2$, se llama clasificación binaria o *dual-case*. Cuando el número de clases es mayor de 2 entonces se dice clasificación múltiple [77].

La clasificación trata de encontrar una función f en la que se cumple que $y = f(x)$ para todos los datos de entrenamiento y que esta función sirva para predecir las clases de los eventos futuros. Aunque en algunos casos es posible predecir varias clases, para esta investigación se predice únicamente una clase a partir de un conjunto de entradas.

En algunos casos, en lugar de predecir una clase interesa obtener un valor probabilístico, es decir, $P(Y|X)$, donde X son los atributos e Y es la clase. Un ejemplo es cuando se quiere calcular el CTR de un anuncio, no se quiere saber si el usuario hará o no clic en el anuncio sino determinar la probabilidad de que se genere un clic. De esta forma, si tres anunciantes están dispuestos a pagar por un clic la misma cantidad se mostrará el que tenga mayor probabilidad de recibir un clic [78].

Algunos ejemplos de métodos de clasificación son las redes bayesianas, las redes neuronales, el *Support Vector Machine*, el árbol de decisión C5.0 o CART.

La clasificación se utiliza mucho en la vida real. Por ejemplo, para predecir si una mujer está en riesgo de cáncer tras analizar una mamografía [79], para evaluar si un cliente pagará en función de su perfil, para el reconocimiento de números escritos a mano, para conocer si un cliente pagará un préstamo, o si una foto pertenece a un rostro y muchos otros casos.

En función de los resultados, una muestra puede pertenecer a los siguientes tipos:

- $Verdaderos\ Positivos\ (VP)$: El número de muestras que fueron correctamente predichas como pertenecientes a la clase.
- $Verdaderos\ Negativos\ (VN)$: El número de predicciones que se predijeron correctamente como no pertenecientes a la clase.
- $Falsos\ Positivos\ (FP)$: El número de muestras que se predijo erróneamente como pertenecientes a la clase.
- $Falsos\ Negativos\ (FN)$: El número total de muestras que se predijeron erróneamente como no pertenecientes a la clase.

A continuación se definirán las principales métricas utilizadas en clasificación donde N es la variable que representa el número de clases e i representa la clase específica.

- $Precisión$: La precisión (*accuracy*) se puede expresar como la tasa de predicciones correctas sobre todas las predicciones. Es la eficacia general del clasificador y puede calcularse mediante la ecuación 2.11.

$$Precisión = \frac{\sum_{i=1}^{N} Vp_i + Vn_i}{\sum_{i=1}^{N} Vp_i + Fn_i + Fp_i + Vn_i} \tag{2.11}$$

- $Sensibilidad$: Esta variable expresa la eficacia de un clasificador para identificar los casos positivos 2.12.

$$Sensibilidad = \frac{\sum_{i=1}^{N} \frac{Vp_i}{Vp_i + Fn_i}}{N} \tag{2.12}$$

- $Especificidad$: Esta variable expresa cuán eficazmente un clasificador identifica los casos negativos 2.13.

$$Especificidad = \frac{\sum_{i=1}^{N} \frac{Vn_i}{Fp_i + Vn_i}}{N} \tag{2.13}$$

- $Recall$: Es una métrica muy común e indica la relación entre las muestras correctamente clasificadas como positivas entre la suma de las muestras clasificadas correctamente como positivas y las erróneamente clasificados como negativos. Como se muestra en la ecuación 2.14.

$$Precisión = \frac{\sum_{i=1}^{N} \frac{Vp_i}{Vp_i + Fp_i}}{N} \tag{2.14}$$

- $F1\ Score$: El F1-Score es una métrica que indica la relación entre la precisión y la sensibilidad, como se expresa en la ecuación 2.15.

$$F1\ Score = 2 \times \frac{Precisión \times Sensitividad}{Precisión + Sensitividad} \tag{2.15}$$

- $Precisión\ Balanceada$: Esta variable equilibra el desempeño del modelo. Se define como el promedio de sensibilidad y la especificidad expresada en 2.16.

$$Precisión\ Balanceada = \frac{Sensitividad + Especificidad}{2} \tag{2.16}$$

- $ErrorOutofbag$: Es una métrica de medición del error en Random Forest y otros modelos de aprendizaje. OOB es el promedio de error de las predicciones en las muestras de entrenamiento utilizando sólo los árboles que no han sido construidos con dichas muestras.

- $Configuraciones$: Cada método prueba distintas configuraciones con distintos parámetros para hacer modelos más exactos.

- $Tiempo\ (ms)$: Es el tiempo, expresado en milisegundos para construir un modelo. Cuanto menor sea este tiempo y mayor sea la precisión, mejor será el método. El tiempo de construcción no es tan importante como el tiempo de respuesta. Porque los modelos pueden ser construidos sin conexión y el tiempo de respuesta debe ser en tiempo real.

2.2.2.2. Métodos supervisados de regresión

La regresión es igual que la clasificación excepto en que la variable de respuesta es continua. Es decir, se tiene una variable real como entrada $X_i \in R$, y una sola respuesta. Igual que en la clasificación se trata de encontrar la función F, $Y = F(X)$ que estime el valor de $Y \in R$ de la forma más exacta [69].

Los métodos de regresión pueden ser distintos a los métodos de clasificación y algunos de los más frecuentes son *Deep Learning*, *Cubist*, *Random Forest*, Regresión lineal con penalización, los modelos de reglas, Análisis de componentes principales, etc.

Algunos ejemplos de regresión pueden ser: predecir el valor de una acción bursátil, predecir el número de ventas de una compañía para un mes, predecir la velocidad de un auto dadas ciertas condiciones, predecir la temperatura, o el número de litros de lluvia para un mes.

La pérdida logarítmica (*logaritmic loss*) se puede definir como el logaritmo de la función de verosimilitud para una distribución al azar de Bernoulli, como se expresa en la ecuación 2.17.

$$Pérdida\, logarítmica = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} y_{ij}\log(p_{ij}) \tag{2.17}$$

La Raíz del Error Cuadrático Medio (*Root Mean Squared Error*) (RMSE) es una medida del rendimiento que se utiliza mucho en las predicciones El RMSE es la raíz cuadrada del promedio de los errores cuadráticos. El RMSE penaliza severamente los errores más abultados. Se puede expresar mediante la ecuación 2.18.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})^2} \tag{2.18}$$

2.2.3. Métodos no supervisados

En el aprendizaje no supervisado, el objetivo no es aprender una relación entre la entrada y la salida. Pues en estos métodos no hay salida y por lo tanto el objetivo es encontrar las regularidades en la entrada. Lo más frecuente en estos modelos es hacer una estimación de la densidad observando cuáles son los patrones que se producen con más frecuencia. Uno de los objetivos principales de los métodos no supervisados es crear grupos a partir de patrones de entrada, lo que se conoce como *clustering* [80].

El *clustering* consiste en la agrupación de un conjunto de observaciones en subconjuntos de modo que las observaciones en el mismo grupo sean similares en cierto sentido. La agrupación es un método de aprendizaje no supervisado, y una técnica común para el análisis estadístico de los datos utilizados en muchos campos [81].

Una aplicación de *clustering* puede ser segmentar a los clientes de una empresa según ciertos factores. Esto puede servir para ofrecer distintos productos en función del perfil de los clientes. También la empresa puede darse cuenta de cuál es el perfil de sus clientes potenciales y por lo tanto, de enfocar las campañas hacia ese público.

2.2.4. Métodos supervisados *Deep Learning*

Las redes neuronales artificiales profundas (*Deep artificial neural network*) están muy en boga debido a que en los últimos años se han obtenido mejores resultados que con los modelos supervisados tradicionales de ML en muchos problemas de aprendizaje automático.

El famoso método de descenso de gradiente se ha utilizado con frecuencia para minimizar los errores no lineales. Los métodos DL se han ido utilizando desde principios de 1960 hasta el año 2006 [82]. Durante todo este tiempo las arquitecturas DL no han sido populares en las investigaciones de ML debido a que no se obtuvieron buenos resultados. Esto se debió a una mala inicialización de los parámetros que se hacía de manera aleatoria [83].

Desde el 2006 al 2017, el DL se ha convertido en uno de los avances tecnológicos más importantes en ML. El DL ha sido aplicado con gran éxito en muchos campos como la medicina, el reconocimiento de la voz [84], el reconocimiento de imágenes [85], el reconocimiento de patrones [86] y el procesamiento del lenguaje natural [82].

Como en el caso de las redes neuronales, DL intenta emular el pensamiento cognitivo del cerebro humano para el tratamiento de la información [87]. Se trata de una familia de algoritmos que tienen arquitecturas formadas en un conjunto de capas que dotan al modelo de un nivel de abstracción mayor.

Los algoritmos DL utilizan gran variedad de técnicas de aprendizaje que están diseñadas para extraer las características de los datos de una manera jerárquica [88]. Es decir, las

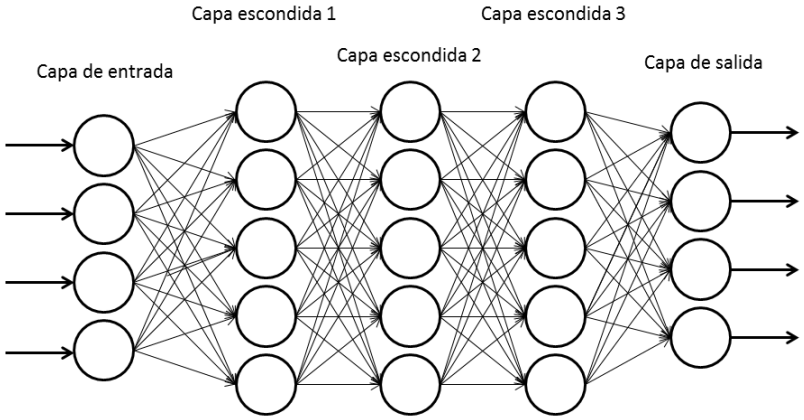

Figura 2.10: Estructura de una red neuronal *Deep Learning*.

características de mayor nivel de abstracción se forman a partir de las características de los niveles inferiores [83]. Esta técnica genera modelos más precisos y con mejores resultados para una amplia variedad de problemas.

Los modelos DL, de la misma forma que las redes neuronales, tienen una arquitectura compuesta por una o varias capas ocultas que permiten múltiples niveles de operaciones no lineales [88]. Esto proporciona a estos modelos una gran capacidad de comprender características que requieren altos niveles de complejidad. DL proporciona unos resultados excelentes en problemas supervisados dentro de la visión en robots [89] y en el reconocimiento de voz [84].

Los métodos DL aplican una metodología basada en crear varios grados de abstracción que se van creando a partir de las entradas. Estos se basan en varias capas de redes neuronales que se entrenan con gradiente estocástico usando la propagación con las capas ocultas. En la Figura 2.10 se muestra una red que tiene cuatro entradas y cuatro salidas, tres capas ocultas con cinco nodos cada una.

En los últimos años se han desarrollado varios métodos de aprendizaje mediante DL. Uno de los avances más importantes en ML es la *Deep Belief Network* (DBN). La DBN es un modelo generativo multinivel en el que cada capa codifica las dependencias estadísticas entre las unidades de la capa inferior. Está diseñado para maximizar el rendimiento de los datos de entrenamiento, es decir, poder generar modelos eficientes con pocas muestras. Las DBNs se han usado con éxito para modelar estructuras de alto nivel en una amplia variedad de dominios como son los dígitos escritos a mano [34] o el reconocimiento de imágenes [90].

Una aplicación interesante de los métodos DL podría ser detectar el fraude por clic en publicidad *online*. Con el fin de detectar el fraude por clic producido por los editores tram-

posos, las redes publicitarias deben analizar un conjunto de variables como la distribución de las IPs o el número total de visitas. Pero también analizan otros datos más complejos como el tiempo entre los clics o los movimientos del ratón. Teniendo todo esto en cuenta, se considera que DL puede ser adecuado para diseñar una herramienta útil para el análisis de esta información compleja.

Los *click-bots* son otro problema que afrontan las redes publicitarias. Por ejemplo, el *botnet chameleon* ha estafado cerca de 6 millones de dólares al mes durante varios meses a las redes publicitarias haciendo falsos clics [44]. Los *click-bots* son programas que se instalan en los equipos de los usuarios sin su consentimiento y generan clics fraudulentos a fin de perjudicar al ecosistema de publicidad *online*. En lugar de contratar a expertos que busquen patrones o analicen el comportamiento de la actividad de *click-bots*, DL podría hacerlo automáticamente y con gran éxito.

2.2.5. Evaluación de modelos de clasificación y regresión

Para que los datos sean más fiables se suele aplicar el método *Cross-validation* (CV) [91]. La técnica *CV* hace diferentes particiones que posteriormente combina para generar distintos conjuntos de muestras tanto para el entrenamiento como para el test. Por ejemplo, si se usan diez particiones $P = 10$, se elegirán aleatoriamente nueve de ellas para el entrenamiento y se dejará una para el test. Por lo tanto, se crean diez modelos diferentes y la precisión de la configuración será el promedio de la precisión de los diez modelos. Para que los resultados sean más fiables este proceso se puede repetir N veces de forma que se crean $P \times N$ modelos. Una vez elegida la mejor configuración se crea un modelo con todas las muestras.

2.2.5.1. Tests estadísticos

El aprendizaje máquina ha ido adquiriendo un gran protagonismo a lo largo del tiempo y su aplicación se extiende cada vez a más campos. Por este motivo, es importante que los investigadores tengan herramientas para evaluar y entender los resultados de estos métodos.

Cuando las muestras siguen una distribución normal se pueden aplicar los conocidos tests como t-student o distribución de Fisher. Sin embargo, hay muchas situaciones en el que el número de muestras es pequeño y no se puede asumir el principio de normalidad en los datos.

Los tests paramétricos se aplican cuando las poblaciones originales cumplen los criterios de normalidad y de homogeneidad. Estos tests sirven para demostrar las hipótesis sobre algún parámetro.

Los tests paramétricos cumplen estas tres características:

• Pueden comprobar hipótesis de ciertos parámetros ($\rho, \beta, \mu, \sigma, ...$).

• Parten del supuesto de que los datos originales cumplen los requisitos de normalidad y homocedasticidad.

• Los datos se analizan en una escala de medida.

Sea X una variable definida por la función de distribución *F*. Se dice que X es paramétrica si su distribución de probabilidad *F* pertenece a una distribución de dimensión finita indexada mediante el parámetro θ:

$$X \sim F \in \mathcal{F}_\Theta = \left\{ F_\theta : \theta \in \Theta \subseteq \mathbb{R}^k \right\}$$

El problema de estos tests es que en muchos tipos de investigaciones científicas no se pueden aplicar. Debido a que no cumplen con algunos supuestos que en determinadas ciencias son difíciles de cumplir.

Además de los tests paramétricos, existen tests estadísticos que permiten contrastar hipótesis sin que cumplan los supuestos necesarios para los tests paramétricos.

Este otro tipo de test se conoce como tests no paramétrico y no plantean hipótesis sobre los datos sino que comprueban algunas propiedades ordinales o nominales de estos. Estos datos son de distribución libre y no necesitan ningún supuesto para ser analizados. Por lo tanto, podemos llamar como no paramétricos a los datos que no cumplen las características de los paramétricos. Esta propiedad permite identificar y clasificar las muchas técnicas de análisis de datos.

A continuación mostramos un resumen de los tests más utilizados son:

- Pruebas para una muestra: Chi-cuadrado, Binomial, Rachas y Kolmogorov-Smirnov.
- Pruebas para dos muestras independientes: U de Mann-Whitney, Kolmogorov-Smirnov, Reacciones extremas de Moses y Rachas de Wald-Wolfowitz.
- Pruebas para varias muestras independientes: H de Kruskal-Wallis y Mediana.
- Pruebas para dos muestras relacionadas: Wilcoxon, Signos y McNemar.
- Pruebas para varias muestras relacionadas: Friedman, W de Kendall y Q de Cochran.

2.2.6. Los métodos de aprendizaje automático en la publicidad *online*

Los avances en la tecnología informática permiten almacenar grandes cantidades de datos. Imagínense la cantidad de información que podría ser almacenada de los usuarios que navegan en Internet o los movimientos de los usuarios que visitan un sitio web durante todo

el día [62]. Lógicamente, requiere varios terabytes por día almacenar dicha enorme cantidad de información.

ML desarrolla técnicas que permiten a las computadoras aprender. En otras palabras, hace posible que el sistema se adapte a los cambios, que prediga el comportamiento futuro y los procesos de toma de decisiones. ML busca algoritmos que generen modelos tan precisos como sea posible, y al mismo tiempo, que ejecute en poco tiempo y con necesidades de memoria pequeñas. Además puede ser mejorado con características adicionales, resultados interpretables, resistencia al ruido y estabilidad [70].

Los métodos de clasificación de ML se han aplicado con éxito para la detección de *spam*. Guzella, Thiago et al. hacen un resumen de los métodos más usados de ML para detectar *spam* [92]. En 1998, Sahami, M. et al., comenzaron a utilizar el clasificador *Naive Bayes* para filtrar el *spam* [93]. El famoso método *Support Vector Machine* (SVM) también comenzó a ser utilizado con gran éxito en 2002. El rendimiento de estos métodos fue superado por los árboles de decisión mediante representación binaria. Los árboles de decisión fueron capaces de utilizar un gran número de características y por lo tanto no era necesario hacer la selección.

En 2003, algunos autores como Poon, Clark y Koprinska utilizaron con éxito las redes neuronales artificiales para categorizar el correo electrónico y para filtrar los mensajes [94]. Por estas fechas, también fue utilizado con buenos resultados el clasificador k-NN. Más tarde, en 2006, Goodman et al utilizaron un modelo de regresión logística que mejoró el rendimiento de los métodos hasta la fecha actual [95]. Tretyakov hace un repaso integral de las técnicas de *spam* más utilizadas y estos se pueden resumir en: NB, k-NN, ANN y SVM [96].

Los métodos de ML también han sido ampliamente utilizados en la predicción de clics a través del parámetro CTR. Al principio, las estadísticas se utilizaron para determinar la probabilidad de que se generara un clic en base a los clics recolectados [97]. Algunas investigaciones se han centrado en otros aspectos como la visualización de los anuncios con alta probabilidad de CTR dependiendo de las características del usuario [98]. Hay investigaciones con el objetivo de calcular el número máximo de veces que un anuncio puede mostrarse a un usuario en particular [56].

El CTR ha sido predicho utilizando múltiples criterios lineales de programación con regresión (MCLPR), *Support Vector Regression* (SVR) y regresión logística y llegaron a la conclusión de que el MCLPR proporciona los mejores resultados en tareas de segmentación. Otros autores, generaron un modelo de redes bayesianas para determinar el CTR de los anuncios nuevos [99]. Inicialmente, se desarrollaron redes bayesianas utilizando las palabras clave de un dominio dado.

El modelo de regresión GBDT fue utilizado para predecir el CTR en base a la información y el valor semántico de los anuncios relacionados. Yin Dawei desarrolló un modelo basado en los factores contextuales para hacer clic en las búsquedas patrocinadas [100].

Kondakindi et al. aplica a nuevos métodos de ML que se han utilizado ampliamente como la regresión logística y el *Naive Bayes* [101]. Tagami et al. aplican los métodos ML en la predicción del CTR para publicidad contextual [102]. Yoganarasimhan utiliza métodos de clasificación y árboles de regresión (CART) [103]. El CTR también puede ser predicho con exactitud utilizando métodos ML basados en reglas de decisión, como C5.0 [104]. Por último, las redes bayesianas [99] y los árboles [105] también han sido usados para la predicción de CTR en búsqueda patrocinada.

Respecto a la previsión de ventas, ARIMA fue la mejor técnica hasta la llegada de los métodos de *machine learning*. Estos métodos tienen una limitación significativa que consiste en que se basan en métodos lineales [106]. Esto es un problema importante porque la predicción de ventas es un problema multivariable afectado por muchos factores internos y externos, y ARIMA no es capaz de explicar las relaciones no lineales.

2.3. Selección de variables

2.3.1. Definición y caracterización

La selección de variables (*Feature Selection*) (FS) se define como el proceso de eliminar aquellas características de una base de datos que son irrelevantes respecto a la tarea a realizar [107]. El objetivo principal de la selección de variables es determinar un subconjunto de características mínimas de un dominio de un problema y, al mismo tiempo, mantener una alta precisión al representar las características originales.

La selección de variables identifica las características útiles para representar los datos y elimina las no relevantes. Esto simplifica la implementación del propio clasificador de patrones y determina las mejores características para el clasificador. Además, la selección de variables tiende a acelerar la velocidad de procesamiento del clasificador y a mejorar los tiempos de respuesta reduciendo la dimensionalidad de entrada. Los métodos de selección de variables pueden mejorar la calidad de la clasificación en términos de precisión, y facilitan la interpretación de los resultados.

En resumen, la FS facilita la comprensión de los datos, reduce los requisitos de almacenamiento, reduce el tiempo de proceso computacional y, finalmente, reduce la dimensionalidad de los datos para mejorar el rendimiento de la clasificación. La FS cada vez es más importante en diversos campos de la ciencia, la ingeniería y las humanidades, incluyendo

muchas disciplinas como la genómica, las ciencias de la salud, la economía, las finanzas y el aprendizaje máquina [108].

Dependiendo de si el conjunto de entrenamiento está etiquetado o no, los algoritmos de selección de variables se pueden clasificar en supervisados, no supervisados y semisupervisados. La selección de variables no supervisada es un problema de búsqueda menos restringido y sin etiquetas de clase, que depende de las métricas de la calidad de agrupación. La selección de variables semisupervisada utiliza datos etiquetados y no etiquetados para estimar la relevancia de las variables. Los métodos supervisados de selección de variables utilizan datos etiquetados y pueden ser categorizados en modelos de tipo filtro, *wrapper* y embebidos.

Los modelos de tipo filtro separan la selección de variables del aprendizaje de clasificación de modo que el sesgo de un algoritmo de aprendizaje no interactúa con el sesgo de un algoritmo de selección de variables. En el esquema univariante, cada característica se clasifica independientemente del espacio de características, mientras que el esquema multivariante evalúa las características en conjuntos. Por lo tanto, el esquema multivariado es capaz de detectar características redundantes.

Algunas de las ventajas de las técnicas de tipo filtro son que se pueden escalar fácilmente, que son computacionalmente más rápidos, y que son independientes del algoritmo de clasificación. Una desventaja común de los métodos de tipo filtro es que no detectan variables redundantes.

Los métodos *wrapper* utilizan la precisión de un algoritmo de aprendizaje predeterminado para determinar la calidad de las características seleccionadas. Las ventajas de los métodos *wrapper* incluyen la interacción entre la búsqueda de subconjuntos de variables y la selección de modelos, y la capacidad de tener en cuenta las dependencias entre las variables. Un inconveniente común de estas técnicas es que tienen un mayor riesgo de *overfitting* que las técnicas de filtro y que son muy computacionalmente más costosas.

Los métodos embebidos logran el ajuste del modelo y la selección de variables simultáneamente. Primero, incorporan criterios estadísticos, como el modelo de tipo filtro, para seleccionar varios subconjuntos de variables candidatas con cierta cardinalidad. En segundo lugar, eligen el subconjunto con la mayor precisión de clasificación. Los métodos embebidos tienen la ventaja de que incluyen la interacción con el modelo de clasificación, y al mismo tiempo son computacionalmente menos costosos que los métodos de *wrapper*.

En función del tipo de salida, los algoritmos de selección de variables pueden dividirse en algoritmos de evaluación de características o en algoritmos de selección de subconjuntos. El método de selección de variables típicamente consta de cuatro pasos: generación de subconjuntos, evaluación de subconjuntos, criterio de parada y validación de resultados.

En la primera etapa se seleccionan un subconjunto de características candidato basado en una estrategia de búsqueda dada, y se envía, a la segunda etapa, para ser evaluado de acuerdo con un cierto criterio de evaluación. El subconjunto que mejor se ajuste al criterio de evaluación se elegirá entre todos los candidatos que hayan sido evaluados después de que se cumpla el criterio de parada. En el último paso, el subconjunto elegido será validado usando el conocimiento del dominio o mediante un conjunto de validación.

La generación de subconjuntos es una búsqueda heurística en la que cada estado especifica un subconjunto candidato para la evaluación en el espacio de búsqueda. Si el conjunto de variables original contiene un número de características N, entonces el número total de subconjuntos candidatos que se generará será de $O\left(2^N\right)$. En la búsqueda secuencial se pueden aplicar varias funciones heurísticas como: ascenso a colinas (*greedy hill-climbing*) [109], la eliminación secuencial progresiva y regresiva, y la selección bidireccional. Dichas heurísticas se utilizan para reducir la búsqueda sin poner en peligro las posibilidades de encontrar el subconjunto óptimo [107]. Por lo tanto, aunque el orden del espacio de búsqueda sea de $O\left(2^N\right)$, se evalúan menos subconjuntos.

Con la búsqueda exponencial (*branch and bound* y *beam search*), el orden del espacio de búsqueda se reduce a $O\left(2^N\right)$ o menos. En la búsqueda aleatoria aunque el espacio de búsqueda sea de $O\left(2^N\right)$, los métodos normalmente buscan un número de subconjuntos menor que 2^N estableciendo un número máximo de iteraciones posibles.

Una función de evaluación mide la calidad de un subconjunto producido por un procedimiento de generación, y este valor se compara con el mejor anterior. Si se encuentra que es mejor, entonces se reemplaza al mejor subconjunto encontrado. Para las funciones de evaluación en los métodos de tipo filtro se utilizan medidas como la información, la dependencia y la consistencia, mientras que para los métodos *wrapper* se usa la precisión [110].

Los procedimientos de generación y las funciones de evaluación pueden influir en la elección de un criterio de parada. Los criterios de parada basados en procedimientos de generación incluyen si se selecciona un número predefinido de características y si se alcanza un número predefinido de iteraciones.

Los criterios de parada basados en una función de evaluación deciden si la adición o eliminación de cualquier variable produce un subconjunto mejor, o si se obtiene un subconjunto óptimo de acuerdo con alguna función de evaluación. El procedimiento de validación no es propiamente una parte del proceso de selección de variables en sí, pero un método de selección de variables debe ser validado.

Este procedimiento prueba la validez del subconjunto seleccionado realizando diferentes pruebas y comparando los resultados con los resultados previamente establecidos, o con los

resultados de los métodos de selección de variables que compiten utilizando bases de datos artificiales, bases de datos del mundo real o ambos.

En Kumar (2014) se muestra una comparación de los métodos de selección de variables existentes [108]. En Khalili (2011) se realiza un estudio comparativo de los métodos de selección de variables para la mezcla finita de modelos de regresión [111].

2.3.2. Análisis de Componentes Principales

El Análisis de Componentes Principales (*Principal Component Analysis*) (PCA) fue desarrollado inicialmente por Pearson a finales del siglo XIX. Posteriormente, en 1930 Hotelling hace un desarrollo en mayor profundidad [112]. Pearson se hizo famoso al aplicar estas técnicas usando equipos informáticos [113].

El PCA transforma el conjunto original de variables $(x_1, x_2, ..., x_p)$ en un nuevo conjunto de variables $(y_1, y_2, ..., y_p)$. Estos nuevos componentes se llaman valores propios (*eigenvalues*). Estos valores no están correlacionados entre sí, y no tienen repetición ni redundancia. El PCA reduce el número de variables pero a la vez trata de mantener tanta información como sea posible.

Los valores propios son una combinación lineal de las variables anteriores y están ordenadas de mayor a menor según la varianza del conjunto original. En el caso ideal, existen m variables que son combinaciones lineales de las variables originales p.

Todos los componentes se pueden expresar como el producto de una matriz compuesta de los vectores propios, multiplicado por el vector X que contiene las variables originales $(x_1, x_2, ..., x_p)$. Dada una matriz de datos, el método PCA busca representar adecuadamente toda la información con un menor número de variables. Donde, X es la matriz original de datos de dimensiones $n \times p$.

$$y = Ax \tag{2.19}$$

Donde

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_p \end{pmatrix}, A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & a_{2p} \\ \dots & \dots & \dots & \dots \\ a_{p1} & a_{p2} & \dots & a_{pp} \end{pmatrix}, x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_p \end{pmatrix} \tag{2.20}$$

y

$$\begin{aligned} Var(y_1) &= \lambda_1 \\ Var(y_2) &= \lambda_2 \\ &... \\ Var(y_p) &= \lambda_p \end{aligned} \tag{2.21}$$

la matriz de covarianza será

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & ... & ... \\ 0 & 0 & 0 & \lambda_p \end{pmatrix} \tag{2.22}$$

porque $(y_1, y_2, ..., y_p)$ se ha construido como variables no correlacionadas, que son

$$\Lambda = Var(Y) = A'Var(X)A = A\Sigma A \tag{2.23}$$

2.3.3. Ganancia de información

El método de selección de la ganancia evalúa el valor de un atributo por la medición de la relación de ganancia a la clase con la ecuación 2.24. Los árboles de decisión se componen de nodos donde los nodos terminales representan la decisión y los nodos no terminales representan evaluaciones sobre uno o más atributos para recorrer el árbol a otro nodo.

Al principio se usó el algoritmo ID3 para construir los árboles y entonces este algoritmo fue mejorado por el C4.5 [114]. Este algoritmo utiliza el ratio de ganancia de las variables en el modelo para construir el árbol de decisión. Para construir el árbol de decisión se selecciona el atributo con mayor entropía. Para cada valor del atributo seleccionado, se crea una rama diferente. Las muestras se dividen en subgrupos, si el valor de salida para ese valor de atributo es el mismo [114]. Cuando esto sucede se completa el proceso de selección de atributo.

$$GainR(Class,\ Attribute) = \frac{H(Class) - H(Class|\ Attribute)}{H(Attribute)} \tag{2.24}$$

Algunos métodos como el ID3, utilizan sólo el *Information Gain* (IG), pero este método tiene la desventaja de que opta por aquellas entradas que tienen una mayor diversidad de valores, aunque estos no sean útiles para determinar el resultado.

Tanto el algoritmo C4.5 como el J48[12] utilizan la relación de ganancia como medida para seleccionar atributos para formar el árbol de decisión [115]. IG es una medida simétrica, lo que indica que la ganancia de Y después de analizar X es equivalente a la ganancia de X después de analizar Y. Esto se puede expresar como:

$$IG = H(Y) - H(YX) = H(X)\text{-}H(XY) \tag{2.25}$$

Esto evalúa la bondad de un atributo como el cociente entre la ganancia respecto de la clase. La ganancia es grande cuando los datos se extienden uniformemente en el árbol y pequeñas cuando todo el mundo está en la misma rama del árbol. El GR favorece las variables con menos valores [116]. La variable GR se obtiene utilizando la ecuación 2.26. Esta ecuación da los valores en el intervalo [0,1]. De tal manera que se establece una relación entre la salida Y y la entrada X. Cuando el GR tiene el valor de "1" quiere decir que esta variable puede predecir totalmente el valor de Y y cuando el valor es "0" significa que no tiene ninguna relación [116, 117].

$$GR = \frac{IG}{H(X)} \tag{2.26}$$

La entropía es una medida que indica la pureza del conjunto de entrenamiento y se utiliza muy a menudo como una medida de la información. Esta métrica es conocida como la "imprevisibilidad" del sistema. La entropía de X se puede expresar con la ecuación 2.27 donde $P(X)$ es la probabilidad marginal de la variable X. Se usará esta variable para calcular el ratio de ganancia. El GR favorece las variables con menos valores [117]. La valor $H(X)$ se obtiene utilizando la ecuación 2.27.

$$H(X) = -\sum_{x \epsilon X} P(x) log_2 P(x) \tag{2.27}$$

2.3.4. El método RFE

En esta tesis se ha utilizado el método recursivo de eliminación de variables (*Recursive Feature Elimination*) (RFE) [118]. El método RFE es de tipo *wrapper*, es decir, utiliza la precisión de los modelos generados para medir la calidad de la combinación de variables escogida. Para evaluar la precisión del modelo se ha elegido el método supervisado *Random Forest*, ya que generalmente da muy buenos resultados y el tiempo requerido para la construcción de los modelos no es muy alto.

Este modelo sigue el esquema mostrado en el Algoritmo 2.1. Como primer paso, un mo-

[12]El algoritmo J4.8 es una versión del algoritmo C4.5 que implementa el paquete de datos Weka.

delo es construido y evaluado independientemente para cada predictor. Posteriormente, los predictores se ordenan de mayor a menor en cuanto a la exactitud de los modelos generados. Entonces, se generan modelos con las variables v_i en orden descendente hasta llegar a la variable V_n, para un conjunto de N variables donde $i = 1, ..., N$. El primer conjunto tiene el predictor 1, el segundo conjunto tiene los predictores 1 y 2, y el tercer juego tiene los predictores 1, 2 y 3, y así sucesivamente hasta el último predictor. El algoritmo RFE elige V_i que es el conjunto de variables que maximice el resultado esperado.

Algoritmo 2.1 Algoritmo RFE para seleccionar las mejores variables.

```
for Cada combinación de muestras do
    Dividir la bbdd original en distintos bloques de entrenamiento o de pruebas
    Entrenar el modelo con los bloques de entrenamiento
    Predecir la clases los bloques de pruebas
    Calcular la importancia de cada variable y ordernarlas de mejor a peor precisión
    for Cada combinación en s_i ∈ S do
        Guardar las variables más importantes v_i
        Entrenar el modelo usando las muestras de entrenamiento con los predictores S_i
        Predecir la clase de los bloques de pruebas
    end for
end for
Calcular el rendimiento del conjunto de predictores sobre S
Determinar el número de predictores apropiados
Estimar la lista final de predictores para mantenerlos en el modelo final
Crear el modelo final basándose en el óptimo S_i usando todas las muestras
```

2.3.5. Métodos basados en Computación Evolutiva

La Computación Evolutiva (CE) es una de las técnicas más ampliamente utilizadas para la función de selección. La CE hace uso de una metáfora de la evolución natural. Según esta metáfora, un problema desempeña el rol de un entorno en el que vive una población de individuos y cada uno representa una posible solución del problema.

El grado de adaptación de cada individuo a su ambiente se expresa con una métrica conocida como función de aptitud (*fitness*). Igual que la evolución en la naturaleza, los algoritmos evolutivos tienen el potencial para producir poco a poco mejores soluciones al problema.

Los algoritmos comienzan con una población inicial de soluciones al azar y, en cada iteración, los mejores individuos son seleccionados y se combinan mediante operadores de variación tales como el cruce y la mutación para construir la siguiente generación. Este proceso se repite hasta que se alcance el criterio de parada, que normalmente consiste en un

número de ejecuciones.

El uso de los algoritmos genéticos (AG) para la selección de variables en el diseño de clasificadores con patrones automáticos fue introducido en 1989 [119]. Desde entonces, los AGs han llegado a ser considerados como una poderosa herramienta para la selección de las características en el aprendizaje automático [120] y numerosos autores lo han propuesto como una estrategia de búsqueda para los métodos de tipo filtro [121], para los métodos tipo *wrapper* [122], para los modelos integrados [123], para los algoritmos con pesos para las características [124], y para los algoritmos de selección de un subconjunto [125]. En [126] se muestra un resumen de las técnicas evolutivas para la selección de variables.

Una de las tendencias actuales en la selección de variables es la optimización multiobjetivo (MO). Un problema recurrente de este enfoque es el conflicto intrínseco entre los dos objetivos del problema: maximizar la exactitud del modelo y reducir al mínimo el número de características.

Los algoritmos evolutivos multiobjetivo [127, 128] (MOEA) han demostrado que son muy eficaces en la búsquedas de las soluciones óptimas a los problemas de MO. Un problema de MO está formulado como un conjunto de problemas de minimización/maximización de una tupla de n funciones objetivo $f_1(\vec{x}), \ldots, f_n(\vec{x})$ donde $\vec{x}$ es un vector compuesto por parámetros que pertenecen a un dominio dado.

Un conjunto $\mathcal{F}$ de soluciones para un problema MO es no dominante (o optimizable según Pareto) si y solamente si para cada $\vec{x} \in \mathcal{F}$, no existe ninguna $\vec{y} \in \mathcal{F}$ para las que se cumple que i ($1 \leq i \leq n$) de forma que $f_i(\vec{y})$ y $f_i(\vec{x})$ para cada j, ($1 \leq j \leq n$, $j \neq i$), $f_j(\vec{x})$ no mejora $f_j(\vec{y})$.

Los algoritmos de tipo MOEAs son especialmente adecuados para la optimización multiobjetivo en la búsqueda de múltiples soluciones óptimas en paralelo y son capaces de encontrar un conjunto de soluciones óptimas para su población final con una sola ejecución.

Una vez que el conjunto de soluciones óptimas se encuentra disponible se puede elegir la más satisfactoria mediante la aplicación del criterio de preferencia. Por lo tanto, el objetivo de un algoritmo multiobjetivo de búsqueda es descubrir una familia de soluciones que sean una buena aproximación para Pareto.

En el caso de la función de selección multiobjetivo, cada solución de la parte delantera representa un subconjunto de características asociado a un *trade-off* entre, por ejemplo, la precisión y la complejidad del modelo. Un *trade-off* significa que se debe perder una cualidad para ganar otra.

El primer enfoque evolutivo de las funciones MO propuesto tenía tres criterios: la precisión, el número de funciones y el número de instancias [129]. Bajo este enfoque, los tres criterios se agregan de uno en uno y, por último, se utiliza un solo objetivo. En [130] se pro-

pone una fórmula para la selección de variables para los problemas de optimización multiobjetivo y también se propone un método de tipo *wrapper* basado en *neuro-fuzzy*, por el cual el algoritmo multiobjetivo genético propuesto en [131] se aplica a problemas de regresión.

En [132] se propone un método de tipo *wrapper* para resolver un problema de optimización, donde el primer objetivo es la precisión mediante un clasificador basado en reglas *fuzzy* y el segundo consiste en aplicar el MOEA que se muestra en [133] y que es una medida agregada de cardinalidad y de granularidad media del subconjunto selección mediante el MOEA.

En [132] se propone un método de *wrapper* modificado que utiliza NSGA para reducir al mínimo el número de características y la tasa de error mediante un clasificador basado en redes neuronales que se aplica al reconocimiento de dígitos escritos a mano.

El método *wrapper* propuesto en [134] considera la precisión del clasificador, la diferencia de la tasa de error entre las clases, y el tamaño del subconjunto mediante el método MOEA donde se propone una técnica basada en una función de adaptación que aplica una penalización para conseguir conservar la diversidad de la población.

El método *wrapper* minimiza la tasa de error y el tamaño del árbol mediante el algoritmo de clasificación C4.5 [135]. También se propone un algoritmo MOEA basado en el concepto de posición dominante y torneo por selección en la que se usa el criterio de desempate cuando ninguno de los individuos domina al resto.

En [136] se propone un método *wrapper* para maximizar la precisión por *cross-validation* en el conjunto de entrenamiento, y para maximizar la precisión de la clasificación en el juego de pruebas, y para reducir al mínimo la cardinalidad de subconjuntos mediante *Support Vector Machine* aplicado al plegamiento de proteínas.

El método NSGA-II se ha modificado para hacer frente a los problemas que tienen el mismo tamaño de características seleccionadas y la misma precisión en la clasificación [137]. También se propuso un algoritmo que combina la optimización evolutiva multiobjetivo con el algoritmo bayesiano utilizando NSGA para minimizar el error y el número de características [138].

NSGA-II se utiliza para reducir la tasa de falsos positivos, la tasa de falsos negativos, y el número de *Support Vectors* para reducir la complejidad computacional. El algoritmo MOGA 3-Objetivos optimiza la sensibilidad, la especificidad y el número de genes, y el algoritmo MOGA 2-Objetivos optimiza la precisión y el número de genes. NSGA-II se utiliza como estrategia de búsqueda y SVM se utiliza para la clasificación.

Algunas investigaciones presentan un filtro de búsqueda local integrado multiobjetivo con un algoritmo mimético que es el producto de una sinergia entre el MOEA y el filtro basado en búsqueda local para la identificación simultánea de clase completa y clase parcial,

donde se utiliza NSGA-II en lugar de MOEA [139]. El enfoque del filtro propuesto incluye métricas de coherencia, de dependencia, de distancia y de información, y NSGA-II se utiliza como estrategia de búsqueda [140].

Los autores llegaron a la conclusión de que las combinaciones entre (*Inter Class Distance + Attribute Class Correlation*), (*Inter Class Distance + Laplacian Score*), y *(Inconsistent Example Pairs + Laplacian Score)* son las mejores. Algunos autores proponen un método *wrapper* de reconocimiento de entidades para maximizar la precisión y la sensibilidad de los modelos basados en la máxima entropía, utilizando NSGA-II en lugar de MOEA [141].

Otros autores introdujeron una modificación en la relación del dominio para el tratamiento de problemas con un gran número de objetivos. Los algoritmos de clasificación utilizan NSGA-II y regresión logística y *Naive Bayes* con corrección de Laplace [142].

En algunas investigaciones, la intención es reducir al máximo el número de características necesarias y a la vez maximizar la precisión del clasificador y/o reducir al mínimo los errores obtenidos mediante SVM para el diagnóstico cardíaco [143].

Algunos autores utilizan el algoritmo genético elitista con reducción del conjunto de pareto (RPSGAe) [144]. Otros aplican la selección de variables a problemas de clasificación mediante un algoritmo multiobjetivo bayesiano artificial con sistema inmune (MOBAIS), con el objetivo de reducir al mínimo los errores de clasificación y la cardinalidad del subconjunto de funciones [145].

El operador tradicional de mutación se sustituye por un modelo probabilístico que representa la distribución de la probabilidad con las mejores soluciones encontradas hasta el momento.

En algunas investigaciones se utiliza SMS-EMOA para el reconocimiento de géneros musicales y estilos con dos combinaciones: la optimización de la sensibilidad y de la especificidad, y la optimización de la precisión y el ratio de las características seleccionadas [146].

Jara, A. et al. proponen un método *wrapper* aplicado a la mortalidad de las infecciones en quemaduras graves para maximizar la precisión de los clasificadores J48 y reducir al mínimo la cardinalidad del subconjunto mediante NSGA-II [147].

Krishna, B. et al. propone una metodología de tipo *wrapper* para optimizar la tasa de error en la minería de datos y también para optimizar el tamaño del árbol de decisiones construido por el algoritmo CART y el tamaño de la red construida por FFNN, utilizando un algoritmo elitista basado en MOGA y NSGA [148, 128].

Karshenas, H. et al. proponen una estimación multiobjetivo del algoritmo de distribución para la selección de variables de un subconjunto que se basa en modelar la unión de los objetivos y las variables, llamada red Bayesiana multidimensional basada en la estimación

del algoritmo de distribución MBN-EDA [149].

Los autores utilizan seis métricas de desempeño diferentes de los clasificadores basados en la precisión de la clasificación, dada por la matriz de confusión y las probabilidades de las clases, y adoptar un enfoque de tipo *wrapper* para evaluar los subconjuntos de características mediante *Naive Bayes* y clasificadores *Naive Bayes* con árboles aumentados. Pati, S. et al. utilizan y se propone un método MOEA donde las poblaciones son generadas por células autómatas híbridas no lineales, y se definen funciones de adaptación, utilizando la aproximación del límite inferior establecido por la teoría *rough set*[13] y la otra con el método de divergencia Kullbak-Leibler [150].

La metodología propuesta hace uso de la autoadaptación, aplicando el algoritmo de selección de variables y al mismo tiempo optimiza los parámetros del clasificador SVM. Recientemente, Kimovski, D. utilizan un enfoque de optimización multiobjetivo paralelo que fue propuesto para abordar problemas de selección de variables con alta dimensionalidad [151].

Se proponen varias alternativas paralelas multiobjetivo evolutivas y los experimentos se evalúan mediante referencias sintéticas y *Brain Computer Interface* (BCI).

2.3.5.1. Los algoritmos evolutivos multiobjetivo ENORA y NSGA-II

ENORA (*Evolutionary Non-dominated Radial slots based Algorithm*) es un algoritmo elitista evolutivo multiobjetivo basado en Pareto que se ha utilizado para una optimización multiobjetivo con restricciones en parámetros reales [152] y para la clasificación de tipo difusa en la predicción de supervivencia [153].

ENORA utiliza una función de supervivencia $(\mu + \lambda)$, donde μ corresponde al tamaño de la población *popsize* y λ se refiere al número de semillas creadas. Fue desarrollada originalmente como una estrategia evolutiva, utilizando una función de selección, con una mutación adaptativa y con una población de tamaño uno, conocido como$(1 + 1) - ES$ [154].

Las nuevas combinaciones y las poblaciones con más de un individuo se introducen más adelante [155]. La técnica $(\mu + \lambda)$ permite encontrar la mejor μ para que sobrevivan los hijos y los padres y es, por tanto, un método elitista.

ENORA utiliza una función de supervivencia $(\mu + \lambda)$ con $\mu = \lambda = popsize$, también utiliza la selección por torneo binario y cruce autoadaptativo y una mutación para optimización evolutiva multiobjetivo. El Algoritmo 2.2 implementa una estrategia $(\mu + \lambda)$ para la optimización multiobjetivo. El algoritmo comienza con la inicialización y la evaluación de una población P compuesta por N individuos.

[13] Es una metodología que utiliza los hechos pasados para crear reglas que modulen la experiencia acumulada.

Algoritmo 2.2 $(\mu + \lambda)$ Estrategia para la optimización multiobjetivo.

Entrada: $T > 1$ ▷ Número de ejecuciones
Entrada: $N > 1$ ▷ Número de individuos en la población
1: Inicializar P con N individuos
2: Evaluar todos los individuos de P
3: $t \leftarrow 0$
4: **while** $t < T$ **do**
5: $Q \leftarrow \emptyset$
6: $i \leftarrow 0$
7: **while** $i < N$ **do**
8: $Padre1 \leftarrow$ Selección de torneo binaria de P
9: $Padre2 \leftarrow$ Selecciónde torneo binaria de P
10: $Hijo1, Hijo2 \leftarrow$ Variación autoadaptativa $Padre1, Padre2$
11: Evaluar $Hijo1$
12: Evaluar $Hijo2$
13: $Q \leftarrow Q \bigcup \{Hijo1, Hijo2\}$
14: $i \leftarrow i + 2$
15: **end while**
16: $R \leftarrow P \bigcup Q$
17: $P \leftarrow N$ Mejores individuos de R según la función de mejor Ranking de población en la población R
18: $t \leftarrow t + 1$
19: **end while**
20: **return** Individuos no dominantes de P

Para cada una de las T generaciones se seleccionan dos padres mediante un torneo binario en la población P (Algoritmo 2.3). Este algoritmo de selección devuelve el mejor entre dos individuos escogidos al azar según la función mejor *Ranking de población* (Algoritmo 2.4). Con esta función, un individuo I es mejor que un individuo J si su ranking es mejor (más bajo) que el ranking del individuo J en la población P. El ranking de un individuo I en una población P, $rank\,(P, I)$, es el nivel no dominante del individuo I entre los individuos J de la población P de forma que $slot\,(I)$=$slot\,(J)$, donde la función *slot* se calcula según la ecuaciones 2.28 y 2.29 donde $d = \left\lfloor \sqrt[n-1]{N} \right\rfloor$ y h_j^I es el objetivo de la función f_j^I normalizada en $[0, 1]$.

$$slot\,(I) = \sum_{j=1}^{n-1} d^{j-1} \lfloor d \frac{\alpha_j^I}{\pi/2} \rfloor \tag{2.28}$$

$$\alpha_j^I = \begin{cases} \frac{\pi}{2} & if\ h_j^I = 0 \\ \arctan(\frac{h_{j+1}^I}{h_j^I}) & if\ h_j^I \neq 0 \end{cases} \tag{2.29}$$

Si dos individuos I y J tienen el mismo ranking, el mejor es el que tenga mayor distancia de población en la parte delantera del individuo. La distancia de población se representa como P^I y P^J respectivamente en la línea 7 del Algoritmo 2.3. La distancia de población de un individuo en una población P es una medida del espacio de búsqueda alrededor del individuo que no está ocupado por ningún otro individuo en la población P.

Algoritmo 2.3 Selección mediante torneo binario.

Entrada: P ▷ Población

1: $I \leftarrow$ Selección aleatoria de P
2: $J \leftarrow$ Selección aleatoria de P
3: **if** I es mejor que J según la función Mejor Ranking de población en la población P **then**
4: **return** I
5: **else**
6: **return** J
7: **end if**

Esta métrica sirve como una estimación del perímetro del rectángulo formado por los vecinos más cercanos como vértices. La distancia *población* se calcula con la ecuación 2.30.

$$Distancia\_población\,(P,I) = \begin{cases} \infty,\ if\ f_j^I = f_j^{max}\ or\ f_j^I = f_j^{min}\ para\ cualquier\ j \\ \\ \sum_{j=1}^{n} \dfrac{f_j^{sup_j^I} - f_j^{inf_j^I}}{f_j^{max} - f_j^{min}},\ en\ otro\ caso \end{cases} \tag{2.30}$$

donde $f_j^{max} = \max_{I \in P} \left\{ f_j^I \right\}$, $f_j^{min} = \min_{I \in P} \left\{ f_j^I \right\}$, $f_j^{sup_j^I}$ es el valor de la objetivo j-ésimo para el individuo adyacente más alto en el objetivo j-ésimo del individuo I y $f_j^{inf_j^I}$ es el valor del objetivo j-ésimo para el individuo inferior adyacente con el objetivo de j-ésimo al individuo I.

El par de padres seleccionado se cruza, se muta, se evalúa y se añade a una población auxiliar inicialmente vacía Q. Este proceso se repite hasta que Q contiene un número de individuos N. Una población auxiliar R se obtiene mediante la unión de las poblaciones P y Q. A continuación, se calcula el ranking de todos los individuos de la población R. Por último, los mejores individuos N de R según la función mejor *Ranking de población* (Algoritmo 2.4) sobrevivirán a la siguiente generación.

Algoritmo 2.4 Función para el mejor Ranking de población.

Entrada: P ▷ Población
Entrada: I, J ▷ Individuos para comparar

1: **if** $rank\,(P,I) < Ranking\,(P,J)$ **then**
2: **return** $Verdad$
3: **end if**
4: **if** $Ranking\,(P,J) < rank\,(P,I)$ **then**
5: **return** $Falso$
6: **end if**
7: **return** $Distancia\_población\left(P^I, I\right) > Distancia\_población\left(P^J, J\right)$

El NSGA-II es un algoritmo evolutivo elitista multiobjetivo basado en Pareto que mejora el algoritmo NSGA anterior mediante la incorporación de una técnica de diversidad explícita [128]. Es quizás el más utilizado que se ha descrito en la literatura.

NSGA-II utiliza, como ENORA, una estrategia $\mu\lambda$ (Algoritmo 2.2) con un torneo binario de selección y una función *Ranking de población.* La diferencia entre NSGA-II y ENORA es cómo se realiza el cálculo de clasificación de los individuos de la población. En ENORA, el ranking de un individuo en una población es el nivel no dominante del individuo en su *slot*, mientras que en NSGA-II el ranking de un individuo en una población es el nivel no dominante del individuo en el conjunto de la población como se puede apreciar en la Figura 2.11.

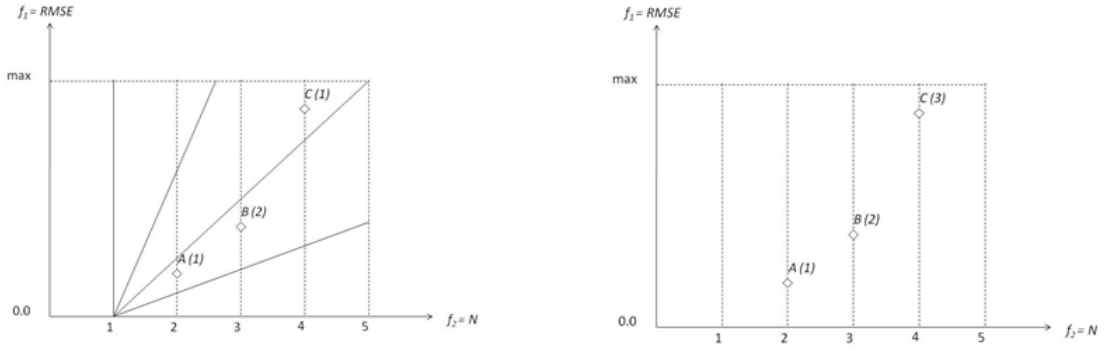

Figura 2.11: *Ranking* de los individuos con ENORA (izda.) contra NSGA-II (dcha.).

La función de selección del método consiste generalmente de cuatro pasos básicos: un subconjunto de generación, un subconjunto de evaluación, un criterio de parada y un resultado validación. La generación del subconjunto es una búsqueda heurística en la que en cada estado se establece un subconjunto de candidatos para la evaluación en el espacio de búsqueda. El número total de los subconjuntos que se generan es $O\left(2^N\right)$, donde N es el número de características.

Algunos ejemplos de generación de subconjuntos mediante el enfoque ascenso de colinas (*Hill-climbing*) son selección secuencial hacia adelante, eliminación secuencial hacia atrás, selección bidireccional, ramificación y poda (*Branch and Bound*), los algoritmos Las Vegas, los algoritmos genéticos, los algoritmos evolutivos y los algoritmos de partículas de enjambre con optimización [109].

El subconjunto de evaluación mide la calidad de un subconjunto producido por un procedimiento de generación de subconjuntos. Algunos ejemplos de las métricas utilizadas para la evaluación de los subconjuntos de los métodos de tipo filtro son la distancia, la información, la incertidumbre, la dependencia y la coherencia mientras que los métodos *wrapper* utilizan la precisión (*accuracy*) [156].

El criterio de parada establece cuando termina el proceso de selección de variables. Este criterio decide si añadir o eliminar alguna característica para producir un subconjunto mejor para un número determinado de ejecuciones.

El proceso de validación comprueba la validez del subconjunto seleccionado mediante la realización de diferentes pruebas. En [157] se muestra un estudio reciente sobre las categorías y un estudio comparativo de los métodos existentes para la selección de variables.

2.3.6. Los algoritmos genéticos

Los algoritmos genéticos fueron desarrollados por John Holland en 1962, investigador de la Universidad de Michigan [158]. Estos algoritmos están basados en la teoría evolutiva del famoso biólogo Charles Robert Darwin (1809-1882). Estos se basan en el famoso principio de "selección natural" para resolver problemas de optimización.

Como se aprecia en la naturaleza los animales compiten constantemente por la comida, el agua o el lugar donde residen. Los mejores individuos de cada especie tienen mayores probabilidades de sobrevivir y por lo tanto, de reproducirse. Y los peores individuos suelen morir por inanición o por ser presas de otros individuos de otras especies.

Darwin aseguró que la supervivencia de una especie se mantenía mediante el proceso de reproducción, cruce y mutación. Esta idea de Darwin se adaptó a la computación mediante algoritmos genéticos para solucionar problemas de optimización [159].

Estos algoritmos tienen una población inicial que se crea de manera aleatoria. A partir de estos individuos se seleccionan los mejores individuos y posteriormente se reproducen entre ellos. Para multarlos o obtener la siguiente población que supuestamente están mejor adaptados que la anterior. Los algoritmos genéticos están orientados a problemas cuyo objetivo es encontrar el valor para un número de parámetros N que maximizan la función de adaptación [160].

Los pasos de un algoritmo genético son evaluar los cromosomas generados, reproducir aquellos cromosomas catalogados como los mejores dándoles mayor probabilidad de reproducirse, mutar algunos genes al azar de algún individuo de la nueva población y crear la nueva población. En el Algoritmo 2.5 se muestra un esquema básico de los AG.

Algoritmo 2.5 Esquema básico de un algoritmo genético.

```
1: Programa principal
2: t ← 0
3: Inicializar P(t);
4: Evaluar P(t);
5: while (¬Termina) do
6:     t ← (t + 1)
7:     P(t) ← seleccionaP(t − 1);
8:     Recombinar P(t);
9:     Mutar P(t);
10:    Evaluar P(t);
11: end while
12: Fin Programa
```

Los principales parámetros de los algoritmos genéticos son la probabilidad de cruce, el tamaño de la población y la probabilidad de mutación. Cada algoritmo genético puede elegir

distintas operaciones de mutación, de selección o de selección [161].

Los operadores que se utilizan en los algoritmos genéticos son selección, cruce, copia y mutación. Los operadores de selección eligen qué individuos sobreviven y cuáles no para dar paso a la siguiente generación. Los operadores de selección más habituales son de torneo en el que compiten entre ellos y los de ruleta en el que se asigna un porcentaje de un valor a cada individuo según su bondad de forma que la suma de todos sea ese valor.

Los operadores de selección de individuos se combinan entre ellos para dar lugar a la siguiente generación. Si siempre se elimina a los padres la estrategia se llama destructiva, pero si solamente se elimina a los padres cuando los descendientes son mejores que ellos entonces se llaman no destructivos.

El operador de copia es una estrategia muy sencilla que consiste simplemente en copiar un individuo de la generación anterior para dar paso a la nueva. El elitismo es una estrategia del operador copia que se basa en copiar siempre a los mejores individuos. Por último, el operador mutación consiste en mutar alguno de los genes de los individuos de forma aleatoria para ver si este nuevo individuo es más apto que el anterior.

Capítulo 3

Viabilidad de la colaboración entre redes para aumentar el rendimiento económico y mejorar la detección de fraude

3.1. Introducción

En los últimos años ha habido un aumento del número de compañías que desarrollan su negocio a través de Internet. Estas compañías ofrecen productos y servicios a un mercado global. Para cualquier empresa que pretenda incrementar sus ventas a través de Internet es fundamental darse a conocer, y la forma más eficaz es mediante la publicidad *online*.

Conforme ha ido pasando el tiempo los anunciantes han sido cada vez más exigentes con los requisitos necesarios para llegar a un público cada vez más específico. Los anunciantes pueden segmentar sus campañas utilizando varios atributos tales como ciudad, tiempo, género, palabras clave, dispositivo o sistema operativo. Esto se conoce como *microtargeting* y reduce el número de visitas que pueden cumplir con los requisitos de los anunciantes, pero a su vez hace que estas visitas tengan un mayor valor [28].

El *Microtargeting* consiste en segmentar a los usuarios según varios atributos para ser dirigidos hacia un pequeño grupo con intereses. Hacer esto asegura que los anuncios se muestren a los usuarios con los requisitos establecidos por los anunciantes y por lo tanto, que sean más propensos a comprar el producto.

Las redes publicitarias pequeñas no pueden ofrecer dichas campañas específicas debido a que no reciben suficientes visitas, y a que sólo una pequeña parte cumple realmente con los requisitos de los anunciantes.

Por esta razón, se ha convertido en algo necesario para las pequeñas redes trabajar juntas para crear un gran mercado mundial de intercambio de anuncios. Cada red está conformada

por un grupo de anunciantes y un pequeño grupo de editores. Para gestionar este cambio se tienen que tener en cuenta ciertas tareas como: la entrega de la factura, la privacidad, la política y la lucha contra el fraude.

En este capítulo se analiza el rendimiento económico de las redes cuando estas operan de manera independiente y cuando colaboran entre ellas. Aunque ya existen modelos en los que las redes colaboran, se han ejecutado algunos experimentos para comprobar y analizar la mejora que obtienen las redes en cada uno de los escenarios. Se han probado escenarios en los que colaboran 2, 3, 4, 5, 10, 25, 50 y 100 redes. Además, se han probado varios algoritmos para ver cuál es el más rápido en ejecución. También se han realizado experimentos en los que las redes comparten información del fraude para visualizar cuanto mejora su capacidad cuando estas colaboran.

La tarea más importante consiste en seleccionar el mejor anuncio entre todos los posibles candidatos y en hacerlo en el menor tiempo posible. Algunos estudios se centran en los diversos factores que deben cumplir los algoritmos de intercambio de anuncios [40]. También hay investigaciones orientadas a optimizar el precio del anuncio o en seleccionar el mejor candidato basándose en una serie de parámetros mediante el uso de fórmulas matemáticas complejas [162].

Para comprobar el beneficio de la colaboración entre redes se han desarrollado diversas soluciones. Se han asignado valores aleatorios a la calidad asumiendo que dichos valores ya se calcularon. En primer lugar, se aplican hilos mediante el entorno de programación C#, que permite ejecutar múltiples hilos simultáneamente.

También se han utilizado matrices de similitud para mostrar anuncios que no cumplen totalmente con los requisitos del anunciante pero que sí tienen un alto grado de similitud. Se han propuesto otras soluciones donde se crea una estructura de tipo árbol con el fin de reducir el número de comparaciones para asignar un anuncio a una red. Para ello, se codifican los valores mediante un *Hash* y se crean árboles AVL y árboles de múltiples nodos.

Estas estructuras hacen posible crear ramas y por lo tanto, mejorar la eficiencia del algoritmo. También se desarrolla un algoritmo mediante el lenguaje Pig Latin de Apache Hadoop. Esta plataforma tiene las herramientas necesarias para el tratamiento de *Big Data* de forma eficaz y sencilla. Para cada algoritmo se ha creado una tabla de resultados y posteriormente se han comparado. Finalmente, se ha llegado a algunas conclusiones y se ha propuesto una serie de mejoras para el futuro.

Hora	Navegador	Versión Nav.	SO	Versión SO	Parámetro N...	Puntu. Calidad	CPC
3	Chrome	20.0.1132.47	Macintosh	Intel 10.5	...	0,634	1,695
14	Chrome	22.0.1229.94	Windows	XP	...	0,982	6,088
15	I. Explorer	01/08/00	Windows	XP	...	0,796	9,37
1	I. Explorer	01/07/00	Windows	XP	...	0,73	6,856
7	Chrome	22.0.1201.0	Windows	Vista	...	0,545	1,704

Tabla 3.1: Formato de la información almacenada sobre los anunciantes.

3.2. Descripción del problema

El proceso de selección de un anuncio es una tarea bastante compleja debido a que hay millones de anunciantes y a que todos pugnan por mostrar su anuncio cuando un usuario visita la página de un editor. Para seleccionar el mejor anunciante se han de evaluar todos los candidatos y dar una respuesta en un tiempo inferior a un segundo, por lo que es realmente importante diseñar algoritmos eficientes.

El problema que se intenta resolver requiere seleccionar el anuncio más adecuado para cada visita en el menor tiempo posible. Para ello, se deben analizar los requisitos de cada campaña. Cuando varios anunciantes cumplan con dichos requisitos se selecciona el que tenga el rango más alto. El rango es un parámetro que tiene como objetivo obtener los mejores beneficios de la red pero al mismo tiempo mostrar la calidad de un anuncio. El valor de un anuncio se calcula con la ecuación 3.1.

$$Valor\ Anuncio = CPC\ \times\ Puntuación\, Calidad \tag{3.1}$$

Cada plataforma utiliza su propio método para calcular la calidad de un anuncio. Por ejemplo, Google no ha revelado nunca cómo calcula el suyo. En la Tabla 3.1 se muestra el formato de la información almacenada de los usuarios que visitaron los anuncios. Son los mismos parámetros que los anunciantes pero sin los valores CPC y con el $Valor\ Calidad$.

Los valores de cada columna son los siguientes:

1. Hora: Se refiere a la hora del día que se hizo la visita.

2. Navegador: Se refiere al navegador del usuario, los más comunes son Internet Explorer y Google Chrome.

3. Versión del navegador: Este parámetro hace referencia a la versión del navegador. Los navegadores constantemente están recibiendo actualizaciones que aumentan el rendimiento y la seguridad.

4. Sistema operativo: Los más comunes son Windows, Mac o Linux.

5. Versión S.O.: De la misma forma que los navegadores cada sistema operativo tienen su propia versión. Por ejemplo: Windows 7, 8 o Mac OS X Lion.

6. *Flash* versión: Algunos navegadores tienen *flash* instalado. Existen muchas versiones de *flash*.

7. ¿Tiene *flash*?: Indica si el usuario utiliza *flash*.

8. Tasa de bits de la pantalla: Indica el número de bits necesarios para mostrar un píxel, generalmente tienen 32 bits.

9. Resolución: Número de píxeles de anchura y altura de la imagen en pantalla.

10. País: Se puede conocer el país mediante la dirección IP de los usuarios.

11. Ciudad: Además de ver el país de origen, también se puede ver la ciudad específica.

12. Idioma: Indica el idioma del sistema operativo, por ejemplo: en, en-us, etc.

13. Dirección: Se refiere a la URL de la página que el usuario está visitando.

14. Nombre de la red: Hace referencia al nombre de la red utilizada por el usuario.

15. Página de acceso: El acceso es la página visitada antes de realizar la visita. Normalmente provienen de los buscadores, pero también pueden acceder directamente a través de un enlace.

16. Tipo de la visita: El tipo de visita puede ser directa, mediante un motor de búsqueda o a través de cualquier otro tipo de página.

17. CPC: El valor máximo que un anunciante está dispuesto a pagar para que se muestre su anuncio.

18. Calidad: Esto indica la calidad del anuncio y se calcula en función de muchos factores como el número de clics o el comportamiento de los usuarios.

En la Tabla 3.2 se pueden ver las opciones de configuración. Los números de cada columna representan los parámetros seleccionados por los anunciantes. Cada número corresponde a los parámetros descritos anteriormente.

Google utiliza supercomputadoras para resolver algoritmos complejos en una décima de segundo. Por ejemplo, cuando los usuarios buscan en el día 08/09/2015 la palabra clave "Brasil" en `Google.es`, este arroja 3.230.000.000 de resultados.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Opción 1		✓		✓						✓						
Opción 2		✓		✓						✓	✓	✓			✓	
Opción 3	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓			✓	
Opción 4	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓

Tabla 3.2: Parámetros configurados en cada opción.

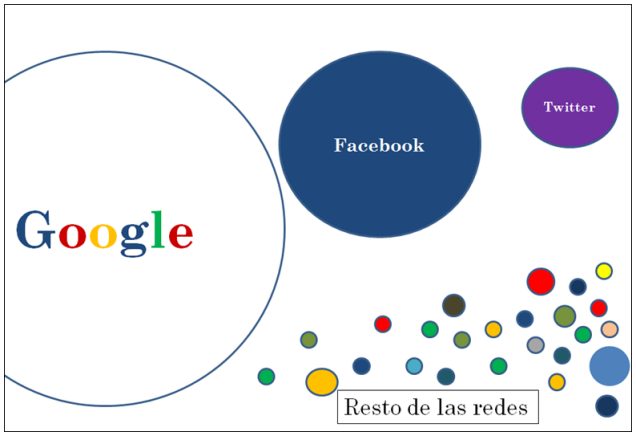

Figura 3.1: Situación actual de las redes publicitarias en la publicidad *online*.

Los principales objetivos de cualquier plataforma de publicidad en Internet son mostrar a los usuarios el anuncio más relevante y reducir a cero el número de fallos en la detección de clics fraudulentos.

La situación de las redes en la publicidad *online* viene reflejada en la Figura 3.1. Las grandes plataformas de publicidad tienen ventajas respecto a las pequeñas porque ofrecen mayor rendimiento a las campañas de los anunciantes y porque tienen sistemas más seguros en la detección de clics fraudulentos [162]. Esto hace que consigan más anunciantes y más editores, lo que les proporciona mayores ingresos creando una espiral en la que las pequeñas redes son cada vez más pequeñas y las grandes cada vez más grandes.

3.2.1. Rendimiento de anuncios

Cuanto más específicos sean los anunciantes en sus campañas menor será el número de páginas en los que se podrán mostrar pero más efectivos serán, por lo que los anunciantes pagarán un mayor precio. Esto se conoce como *microtargeting*.

Para poder desarrollar bien el *microtargeting* se debe filtrar por una serie de parámetros como son las palabras clave por las que se accedió, la edad, el género, el nivel de ingresos, la localización o los gustos que aparecen en el perfil de un usuario.

Hay otra serie de atributos que no influyen tanto como son el navegador, el buscador, el

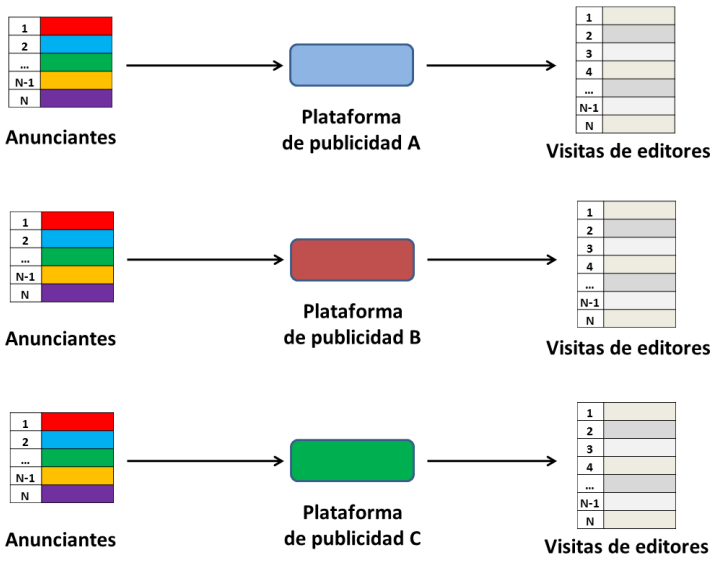

Figura 3.2: Plataformas publicitarias operando de forma independiente.

sistema operativo o el dispositivo pero que también es bueno tener en cuenta. Cuando un usuario que visita una página encaja con las características deseadas por el anunciante se mostrará el anuncio y si se hace clic, se hará un cobro al anunciante.

Una plataforma grande con muchos editores fácilmente encontrará alguna página que esté relacionada con los requisitos de algún anunciante y que tenga buena aceptación. Por contra, si la red publicitaria tiene pocos editores y funciona de manera independiente como se muestra en la Figura 3.2, los anuncios tendrán poca cobertura[1], es decir, se mostrarán pocas veces y no llegarán a muchos usuarios. Para solucionar este problema se tendrán que hacer campañas genéricas, pero esto tiene la desventaja de que tienen un rendimiento menor.

3.2.2. Detección de fraude

Lo primero que se debería responder es si el fraude por clic es realmente una amenaza. En el caso de que los clics fraudulentos representasen el 15 % de los clics totales, como aseguran algunos expertos [163], y que de estos no se lograra detectar el 20 %. Esto quiere decir que se podría descontar ese porcentaje del 3 % $(0,15 \times 0,2)$ del precio que pagan los anunciantes. Según Tuzhilin, estadísticamente es posible que alguno de los anunciantes no esté satisfecho pero si la gran mayoría sí lo están, la plataforma tendrá éxito [20].

[1]La cobertura es un valor entre 0 y 100, y representa la cantidad de veces que se ha mostrado un anuncio a un usuario. Tener una cobertura del 50 % significa que la mitad de las visitas no han sido aprovechadas para obtener ingresos, ya que no cumplía con los requisitos de ningún anunciante.

El problema surge cuando aumenta el número de clics fraudulentos o disminuye la capacidad de detectar clics de una plataforma frente a la competencia. Los editores preferirán trabajar con la plataforma que ofrezca mejor rendimiento por su espacio publicitario y los anunciantes buscarán la que proporcione mejores resultados en sus campañas [164].

Las grandes redes publicitarias hacen inversiones millonarias y suelen contar con equipos especializados que están continuamente mejorando el sistema de detección de fraude. Su capacidad de detectar clics falsos es muy superior al de las pequeñas redes. Google, por ejemplo, conoce el CTR de todas las categorías de páginas de Internet, por lo que si una página tiene estadísticas diferentes al resto será fácilmente detectada.

También tiene a su disposición Gmail, que permite analizar cosas tan complejas como si entre las IPs que han hecho clic ha habido algún tipo de comunicación mediante mensajes o chat. Si Google es vulnerable como ha demostrado la empresa *Spider.io* con el *Chameleon botnet* [44, 45], es fácil imaginarse que las pequeñas plataformas lo serán mucho más.

Según los principios de Kerckhoffs, las principales plataformas deberían publicar las técnicas que utilizan los estafadores y los métodos para detectarlos, de forma que los sistemas fuesen cada vez más seguros. También algunas investigaciones hablan de la conveniencia de colaborar entre las redes para mejorar la detección de fraude [165]. Sin embargo, la gente confía en las grandes redes por su capacidad de detectar clics. Si todas las plataformas fuesen igual de seguras dejaría de ser una ventaja competitiva.

3.3. Colaboración entre pequeñas redes

Algunos autores afirman que el intercambio de anuncios representa el futuro de la publicidad en Internet y la solución a las pequeñas plataformas publicitarias. Pero para que el intercambio de anuncios tenga éxito primeramente tiene que resolver el problema del fraude por clic, las cuestiones legales respecto a la privacidad de los usuarios y desarrollar un modelo de intercambio que genere beneficios a todas las plataformas que participan.

3.3.1. Colaboración para aumentar el rendimiento

El intercambio de anuncios consiste en que las plataformas intercambien las visitas que no cumplen con los requisitos de ninguno de sus anunciantes o sencillamente en que busquen un anunciante dispuesto a pagar un mayor precio como se muestra en la Figura 3.3. En este modelo, los anunciantes compran espacios si cumplen ciertos requisitos y el editor deja un espacio en sus páginas para que se rellene con el anuncio más rentable.

Imaginemos que existen dos pequeñas plataformas de publicidad llamadas SpainOnline97 de España y BrasilMarket43 de Brasil. La mayoría de los anunciantes de SpainOnli-

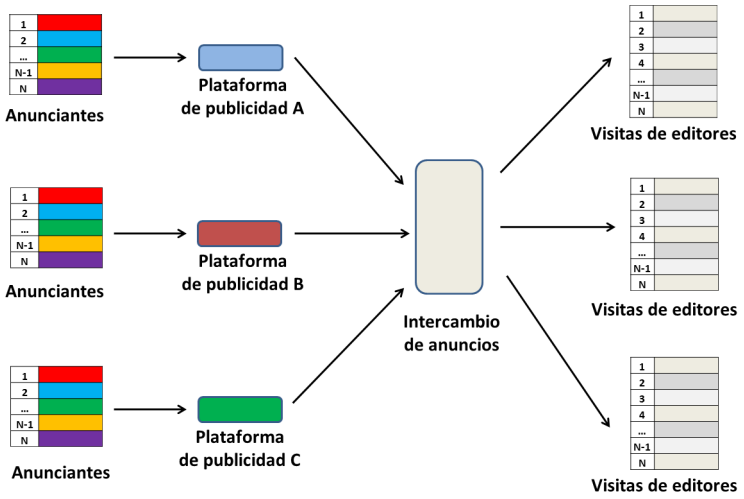

Figura 3.3: Colaboración entre plataformas para el intercambio de anuncios.

ne97 será de habla hispana y la mayoría de los anunciantes de BrasilMarket43 será de habla portuguesa. Cuando un usuario de Brasil visite una página de SpainOnline97 se mostrarán anuncios relacionados con España que al individuo de BrasilMarket43 probablemente no le interesen. Por otro lado, cuando un individuo de España visite páginas de Brasil le aparecerán anuncios relacionados con Brasil que seguramente tampoco le interesan al usuario de España. Si estas dos redes intercambiaran anuncios entre ellas, cuando el individuo de España visite una página de BrasilMarket43 le podrían aparecer anuncios de España, de modo que es más probable que le interese la publicidad y compre el producto. Lo mismo ocurriría con el de Brasil.

La mayoría de las plataformas de anuncios sigue los estándares del IAB, lo que permite hacer intercambios con mayor facilidad. Esto puede parecer sencillo cuando solamente colaboran dos redes, pero cuando lo hacen cientos de redes con miles de anunciantes hay que manejar factores como el volumen de intercambio, la remuneración de cada red, el fraude que pueden cometer los distintos anunciantes o el reparto de las visitas de forma distribuida.

Para hacer un intercambio de anuncios adecuado es necesario desarrollar un algoritmo teniendo en cuenta que el anuncio se debe mostrar en escasas décimas de segundo y que a la vez ha de estar muy bien elegido. Para ello, es recomendable utilizar hilos (*threads*) en la programación de dicho algoritmo, ya que permiten ejecutar varios procesos de forma simultánea y quedarnos con el que obtenga un mejor resultado.

3.3.2. Colaboración para detección de fraude

Una solución para disminuir los costos podría ser que las plataformas de publicidad externalizaran la detección de clics a empresas expertas especializadas. El problema radica

en que estas mismas empresas estarían tentadas de construir nuevas amenazas para asegurar su trabajo o de aliarse con editores tramposos para obtener mayores beneficios.

Existen múltiples amenazas a las que se enfrentan las plataformas de publicidad como son los *click-bots*, el tráfico ilegal o los usuarios malintencionados. Pero estas amenazas son las mismas para todas las plataformas por lo que si una plataforma detectara una IP maliciosa y avisará al resto, esta IP dejará de ser una amenaza [165].

Compartir información es una ventaja para que todas las plataformas puedan dar mejor servicio a los anunciantes y para que el número de clics fraudulentos no detectados sea cada vez menor.

Las ventajas de un modelo colaborativo son:

- Conocer el CTR de páginas de las otras plataformas, de tal manera que si un usuario tiene una página con similares características a otras redes y un CTR muy diferente será sospechoso.
- Compartir IPs sospechosas de ser fraudulentas.
- Actualizar la lista de proxies[2] para invalidar los clics que procedan de estos.
- Compartir métodos para detectar nuevos *click-bots*.
- Contrastar los ratios de un determinado editor con los editores de todas las demás plataformas. Esto dirá si se sale de la norma.
- Calcular el porcentaje de clics fraudulentos para poder aplicar descuentos a nuestros anunciantes.

3.3.3. Respeto a la privacidad

Las plataformas de publicidad pueden recolectar mucha información sobre los usuarios que utilizan sus servicios. Google, Microsoft o Yahoo tienen capacidad para conocer a qué hora se conectan sus usuarios, qué páginas visitan, a qué individuos escriben correos o con qué personas chatean, etc. Esta información la conocen gracias a las *cookies*, la IP, el historial o el registro de sesión.

Google Webmaster tools, que se muestra en la Figura 1.1, y Google Analytics, permiten a Google obtener estadísticas sobre muchos factores como los tiempos en que un usuario

[2]Un proxy es un programa o dispositivo que realiza una conexión a Internet desde otro ordenador. Sirve para tener anonimato o mayor seguridad. En el caso del fraude por clic sirve para hacer un clic sin que se detecte la IP.

permanece en las distintas páginas, el promedio de páginas visitadas, los navegadores más utilizados o los dispositivos desde los que se accede.

Cuanta más información se tenga de los usuarios y mejor se utilice, más personalizados podrán ser los anuncios que se muestren. Pero por otra parte, siempre se tiene que respetar el derecho a la intimidad. Para protegerse frente a posibles denuncias por estos aspectos, las plataformas tienen sus condiciones o políticas que el usuario debe aceptar para acceder a esos servicios.

3.4. Algoritmos para mejorar la gestión de anuncios

3.4.1. Entorno de evaluación

Para resolver el problema se ha utilizado un CPU Intel (r) Core (TM) i5-2400 a 3,10 GHz con 16Gb de RAM corriendo *Windows 7 Pro Service Pack 1 64 bits*. Mediante el uso de este *hardware* se puede resolver el mismo problema de diferentes formas. Se ha desarrollado el código mediante hilos (*threads*), y mediante árboles AVL, así como árboles de varios nodos. Para ello, se ha utilizado Microsoft Visual C# 2010 en el entorno *C# Express*. También se ha utilizado Pig Latin mediante tecnología desarrollada por Yahoo llamado Hadoop.

Para ejecutar Hadoop hay una máquina virtual llamada Sandbox de Hortonworks 2.1 en el sistema operativo Red Hat, que se ejecuta sobre Oracle Virtual Box, 4.3.14 r95030 y todos los elementos antes mencionados.

3.4.2. Aumento de la cobertura de anuncios

En este punto se pretende visualizar cómo aumenta la cobertura de los anuncios, es decir, el porcentaje de visitas que pueden ser satisfechas en función del número de redes que colaboran. Para realizar las pruebas se han obtenido un total de 104.151 visitas reales del historial de una página web llamada buscadoresdeinternet.net desde el 01/06/2012 hasta el 01/01/2013. En la Figura 3.4 se puede ver una imagen de dicha web.

Cada visita tiene una serie de campos entre los que se encuentran: hora, navegador, versión del navegador, sistema operativo, versión del sistema operativo, versión de *flash*, ¿tiene *flash*?, bits de la pantalla, resolución pantalla, país, ciudad, idioma, dirección de red, nombre de la red, página de acceso y tipo de visita.

Las visitas se han exportado a una tabla Excel desde Google Analytics de la forma que se muestra en la Tabla 3.3.

Se ha comparado cada visita del fichero con los requisitos de las campañas de los anunciantes de las X redes que participan, donde $(X = 1, 2, 3, 5, 10, 25, 50, 100)$. Cada una

Figura 3.4: Imagen de la página de buscadoresdeInternet.net (18/12/2015).

	Campo 1	**Campo 2**	**Campo 3**	**Campo...**	**Campo N-1**	**Campo N**
Visita 1	Firefox	16.0	Windows	7	11.4 r402	24-bit
Visita 2	Chrome	22.0.1229.92	Windows	XP	11.4 r31	32-bit
Visita...	I. Explorer	01/08/00	Windows	7	(Sin configurar)	32-bit
Visita N-1	I. Explorer	01/08/00	Windows	7	11.1 r102	32-bit
Visita N	Chrome	21.0.1180.89	Windows	XP	11.3 r31	32-bit

Tabla 3.3: Almacenamiento de las visitas de los usuarios.

	Campo 1	**Campo 2**	**Campo 3**	**Campo...**	**Campo N-1**	**Campo N**
Anunciante 1	Chrome, Firefox	16.0	Windows	XP,7	11.4 r402	24-bit,32-bit
Anunciante 2	Chrome	22.0.1229.92	Windows	XP	11.4 r31	32-bit
Anunciante N	Internet Explorer	01/08/00	Windows	XP,7	(Sin configurar)	32-bit

Tabla 3.4: Formato de almacenamiento de las características de los anunciantes.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Opción 1										✓						
Opción 2		✓								✓						
Opción 3		✓		✓						✓						
Opción 4		✓		✓						✓		✓			✓	
Opción 5		✓		✓						✓	✓	✓			✓	
Opción 6		✓		✓				✓		✓	✓	✓			✓	
Opción 7		✓		✓				✓	✓	✓	✓	✓			✓	
Opción 8	✓	✓		✓				✓	✓	✓	✓	✓			✓	
Opción 9	✓	✓		✓	✓			✓	✓	✓	✓	✓			✓	
Opción 10	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓			✓	
Opción 11	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓			✓	✓
Opción 12	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓
Opción 13	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Tabla 3.5: Parámetros seleccionados por cada una de las opciones.

de las redes tiene 10 campañas, lo que representa un total de 1.000 campañas de forma simultánea.

Los valores que selecciona cada anunciante para cada parámetro se establecen de forma aleatoria basándose en la probabilidad de ocurrencia. Es decir, si el 90 % de los sistemas operativos son de Windows en el historial de visitas la probabilidad de que el anunciante escoja el valor de Windows para el parámetro Sistema operativo será de un 90 %, tal y como se indica en la Tabla 3.4. Los anunciantes seleccionan una serie de campos para mostrar un anuncio y la plataforma de publicidad coloca mediante un algoritmo esos anuncios en las páginas web de los editores.

Para cada una de las opciones se configuran distintos parámetros. Cuanto mayor es el número de la opción mayor es el número de parámetros que se configuran de forma que se hacen campañas más específicas y por lo tanto, más difíciles de cubrir.

En la Tabla 3.5 se puede ver que el eje Y representa el número de opciones y el eje X representa los parámetros que se pueden configurar y que están explicados en la Tabla 3.4. Si la casilla está en blanco indica que no es un requisito en la campaña, en cambio, si está marcado indica que el anunciante exige que ese campo tenga un valor concreto para que se pueda mostrar su anuncio.

Cobertura	1	2	3	5	10	25	50	100
Opción 1	0,65901	0,79542	0,85574	0,9092	0,95288	0,98307	0,99184	0,9943
Opción 2	0,35315	0,51562	0,63524	0,73938	0,8384	0,92112	0,95214	0,97189
Opción 3	0,34136	0,5101	0,60923	0,71234	0,82205	0,8987	0,93521	0,95982
Opción 4	0,15334	0,25803	0,33614	0,43064	0,53901	0,65418	0,73267	0,79566
Opción 5	0,02476	0,04473	0,06103	0,09263	0,15475	0,25647	0,3468	0,44492
Opción 6	0,01351	0,02455	0,03379	0,05396	0,09425	0,17517	0,25696	0,34725
Opción 7	0,00169	0,00344	0,00549	0,00882	0,01675	0,03739	0,06599	0,10764
Opción 8	9,40E-005	0,00019	0,0003	0,0005	0,001	0,00249	0,00487	0,00955
Opción 9	6,90E-005	0,00015	0,00023	0,00036	0,00071	0,00176	0,00343	0,00663
Opción 10	1,90E-005	3,50E-005	5,10E-005	8,30E-005	0,00015	0,00036	0,00073	0,00146
Opción 11	1,20E-005	2,50E-005	4,30E-005	7,40E-005	0,00014	0,00038	0,00074	0,00147
Opción 12	3,00E-006	7,00E-006	9,00E-006	1,40E-005	3,10E-005	7,60E-005	0,00016	0,0003
Opción 13	1,00E-006	2,00E-006	2,00E-006	4,00E-006	0,00001	2,60E-005	5,10E-005	0,0001

Tabla 3.6: Cobertura respecto al número de redes y a las opciones configuradas.

En la Tabla 3.6 se representa la cobertura en función del número de redes que colaboran (Eje X) y las distintas opciones que seleccionan los anunciantes al realizar sus campañas (Eje Y). Los parámetros que se configuran en cada opción vienen indicados en la Tabla 3.5.

Como se puede apreciar en la Tabla 3.6, a medida que colaboran un número mayor de redes la cobertura es mayor. El Eje X representa el número de redes y cuantas más redes colaboran mayor es la cobertura que representa el porcentaje de anuncios que se logra publicar. El eje Y representa la opción elegida por el anunciante. Cuanto mayor es la opción más restrictiva es la opción y por lo tanto menos anuncios logran mostrarse. En cambio, estos anuncios tendrán una mayor aceptación por parte de los usuarios.

3.4.3. Distribución de las visitas

Además de intentar aumentar la cobertura haciendo que participen el resto de las plataformas de publicidad también se pretende repartir las visitas de la forma más equitativa. Para ello, se han desarrollado tres algoritmos y para medir la calidad de dichos algoritmos se ha utilizado como métrica la desviación media. Cuanto menor sea la desviación media mejor será el algoritmo que se utiliza.

La desviación media es la media aritmética de los valores absolutos de las desviaciones respecto a la media y se representa por Dm. Se calcula mediante la ecuación 3.2.

$$Dm = \frac{1}{n} \sum_{i=1}^{N} \mid X_i - X \mid \tag{3.2}$$

El algoritmo llamado Simple consiste en preguntar en primer lugar a la red número 1 y en caso de que no pueda satisfacer la petición, se preguntará a la número 2 y así sucesivamente hasta llegar a la última red. En la Tabla 3.7 se muestran los resultados obtenidos.

Simple	2	3	4	5	10	25	50	100
Opción 1	25.317,5	25.074,4	22.688,6	20.387,7	13.193,3	6.422,08	3.512,67	1.862,84
Opción 2	8.876,84	10.684,2	10.747,8	10.379,5	8.419,45	4.901,6	2.913,74	1.634,75
Opción 3	10.042,8	10.521,1	10.715,2	10.377,8	8.224,26	4.796,97	2.852,07	1.603,25
Opción 4	4.939,88	5.217,37	5.425,4	5.236,4	4.307,7	2.754,35	1.763,37	1.064,59
Opción 5	620,12	776,33	818,67	819,45	816,09	679,08	525,36	374,93
Opción 6	396,52	475,69	499,97	511,62	497,21	428,71	350,36	268,62
Opción 7	55,81	67,19	85,37	90,59	92,42	90,14	86,49	74,27
Opción 8	5,14	5,72	6,04	5,95	6,63	6,81	6,76	6,92
Opción 9	3,44	3,68	4,15	4,02	4,77	5,09	5,18	5,15
Opción 10	1,26	1,4	1,41	1,4	1,45	1,52	1,56	1,6
Opción 11	1,11	1,34	1,44	1,52	1,54	1,59	1,59	1,55
Opción 12	0,24	0,28	0,33	0,34	0,42	0,48	0,47	0,47
Opción 13	0,09	0,14	0,17	0,19	0,21	0,2	0,2	0,19
Total	**276.473**							

Tabla 3.7: Desviación promedio en el algoritmo Simple.

El algoritmo *Round Robin* pregunta en el primer ciclo a la red número 1 en primer lugar pero la segunda vez pregunta en primer lugar a la red número 2. Cada vez que reparte una visita empieza preguntando por la red siguiente de la última red que preguntó. Los resultados obtenidos los se muestran en la Tabla 3.8.

El algoritmo llamado Mínimo siempre pregunta a la red que menos visitas ha cubierto. Para ello, se ayuda de una tabla en la que lleva la cuenta del número de visitas que se han repartido en cada red. Los resultados se muestran en la Tabla 3.9.

Para comparar los resultados se ha realizado la suma de todas las pruebas para cada algoritmo. Cuanto menor sea la suma mejor será el algoritmo. El que mejores resultados proporciona es el algoritmo Mínimo (13.595,04), después el *Round Robin* (77.115,58) y por último el Simple (276.473,42).

3.4.4. Algoritmo para la detección de fraude

Para comprobar la mejora en la detección de fraude en un entorno colaborativo se utiliza la técnica del *captcha* [166] y la de los anuncios irrelevantes [167]. Esto servirá para detectar IPs fraudulentas. La técnica de los anuncios *captcha* consiste en pedir a los usuarios resolver un *captcha* como el que se muestra en la Figura 3.5 cuando quieran acceder al contenido del anuncio. Si se pusiera un *captcha* en cada acceso a un anuncio los usuarios se podrían desesperar por lo que se aplicará solamente el 20 % de las veces.

La técnica de anuncios irrelevantes consiste en mostrar a un determinado usuario anuncios que no relacionados con su perfil. De esta manera los clics que no provienen de un verdadero interés en el producto sino que se ejecutan por motivos maliciosos. El comporta-

Round Robin	2	3	4	5	10	25	50	100
Opción 1	4.747,32	4.666,87	3.856,47	3.486,34	1.925,38	798,02	409,9	207,55
Opción 2	2.981,15	3.936,11	3.509,33	3.172,09	1.916,85	890,94	461,24	233,56
Opción 3	3.951,06	3.976,8	3.721,39	3.456,14	2.009,65	900,78	451,18	230,77
Opción 4	2.646,23	2.643,42	2.668,82	2.349,33	1.599,45	774,59	425,82	224,63
Opción 5	634,12	692,18	659,68	729,41	659,47	445,21	277,43	160,23
Opción 6	390,75	433,59	453,86	456,33	454,81	316,1	221,94	137,84
Opción 7	70,19	78,44	87,07	92,67	95,09	90,39	77,25	61,37
Opción 8	4,37	5,32	5,51	5,54	6,44	7,01	7,05	6,82
Opción 9	3,19	3,71	4,25	4,8	4,85	5,15	5,21	5,21
Opción 10	0,99	1,21	1,28	1,28	1,42	1,55	1,59	1,59
Opción 11	1,24	1,23	1,37	1,41	1,46	1,48	1,5	1,49
Opción 12	0,31	0,39	0,36	0,37	0,4	0,45	0,44	0,45
Opción 13	0,08	0,14	0,16	0,19	0,23	0,2	0,2	0,2
Total	**77.116**							

Tabla 3.8: Desviación promedio del algoritmo *Round Robin*.

Mínimo	2	3	4	5	10	25	50	100
Opción 1	255,76	56,35	22,82	15,74	4,24	0,78	0,43	0,37
Opción 2	1290,72	539,18	287,88	76,91	16,09	1,75	0,61	0,41
Opción 3	679,6	466,18	178,16	96,93	12,12	2,69	0,88	0,43
Opción 4	1.127,77	815,04	630,44	387,4	153,21	16,58	2,85	0,95
Opción 5	634,4	692,12	647,37	625,64	493,62	208,01	74,87	15,33
Opción 6	376,96	442,27	432,25	425,59	344,21	202,55	97,03	32,26
Opción 7	62,33	77,08	81,23	86,55	91,51	81,2	66,91	47,92
Opción 8	5,13	5,68	5,8	5,96	6,42	6,82	6,75	6,65
Opción 9	3,15	4	4,7	4,51	5,1	5,15	5,19	4,93
Opción 10	1,09	1,54	1,51	1,51	1,47	1,55	1,52	1,51
Opción 11	1,17	1,44	1,56	1,53	1,54	1,58	1,61	1,62
Opción 12	0,37	0,4	0,44	0,41	0,43	0,49	0,47	0,47
Opción 13	0,1	0,14	0,16	0,17	0,2	0,2	0,2	0,2
Total	**13.595**							

Tabla 3.9: Desviación promedio del algoritmo Mínimo.

Figura 3.5: Ejemplo de código *captcha* para evitar el *spam*.

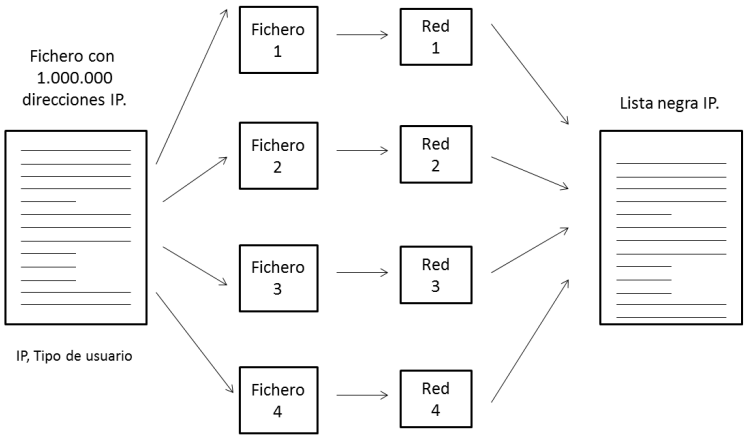

Figura 3.6: Modelo colaborativo entre redes para la reducción de fraude.

miento esperado de un usuario es que no haga clic, por lo que si hay un clic es probable que se trate de un *botnet* o de un grupo de usuarios fraudulentos poco entrenados. Si se abusa de esto, el usuario fraudulento se sentirá vigilado y tendrá una retroalimentación de que algo no va bien y cambiará su comportamiento.

Para comprobar la mejora en la detección de fraude en proporción a las redes que colaboran se diseñará un modelo en el que las redes intercambian las direcciones IP que tengan una alta probabilidad de ser fraudulentas como se muestra en la Figura 3.6.

El experimento consiste en crear un fichero con 100.000 IPs de las cuáles, el 10 % proceden de anuncios irrelevantes. De este 10 %, el 75 % procede de *botnets*, el 20 % de usuarios fraudulentos y el 5 % de usuarios válidos. Del 90 % restante de las visitas, el 80 % procede de usuarios válidos, el 15 % de *botnets* y el 5 % de usuarios fraudulentos. En el experimento colaboran 1.000 redes, de las cuales cada una obtiene 2.000 visitas de forma aleatoria del fichero inicial.

Para medir el rendimiento en la detección de fraude se hace una comprobación mediante *captcha* el 20 % de las veces, salvo en el caso de los anuncios irrelevantes en los que se hace el 100 % de las veces. Cuando un *botnet* no resuelva el *captcha* se añadirá a una lista de IPs sospechosas.

El porcentaje de detección consiste en dividir el número de *botnets* detectados entre el número de *botnets* totales. En el experimento en primer lugar participa una sola red de forma que se encuentra una lista de IPs sospechosas vacía. A medida que van participando más redes, la lista va aumentando el número de IPs sospechosas, de forma que la red número 500 tiene las IPs fraudulentas detectadas por las 499 anteriores. Esto explica el hecho de que conforme aumente el número de redes que se ejecutan su capacidad de detectar IPs fraudulentas es mayor.

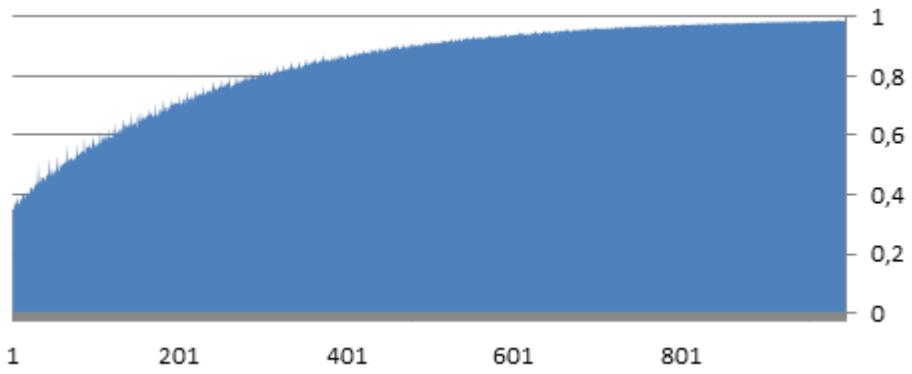

Figura 3.7: IPs detectadas en función del número de redes que colaboran.

En la Figura 3.7 se pueden ver los resultados de cómo mejora la detección de fraude cuantas más redes colaboran entre ellas y el código *captcha* se aplica a más visitantes. El eje X representa el número de redes que colabora en la detección de fraude y el eje Y representa el porcentaje de IPs fraudulentas detectadas.

3.5. Optimización del intercambio de anuncios

3.5.1. Hilos y matrices de similitud en el intercambio de anuncios

Hoy en día se cuenta con procesadores multinúcleo que se pueden interconectar y que permiten analizar grandes volúmenes de información. Los hilos permiten procesar varias instrucciones a la vez, operando sobre el principio de que problemas grandes se pueden dividir en problemas más pequeños que son resueltos simultáneamente.

3.5.1.1. Aplicando hilos al intercambio de anuncios

Los hilos son procesos que pueden ser ejecutados al mismo tiempo. Los distintos hilos de ejecución comparten una serie de recursos tales como espacio en memoria, archivos o claves de autenticación.

Los hilos permiten comparar varios anunciantes de forma simultánea en el tiempo en vez de tener que comparar la visita con cada uno de los anunciantes de forma secuencial. El programa que se ha desarrollado crea 100.000 hilos, siempre y cuando no se ponga un límite de tiempo, que se ejecutan a la vez para cada una de las 1.000 visitas que se analizan. Cada uno de los 100.000 hilos compara 100 campañas almacenadas en un fichero con la visita, lo que hace un total de 1.000.000 de campañas. Cuando finalizan todos los hilos el programa escribe la mejor solución en un fichero.

Los Algoritmos 3.1 y 3.2 se ejecutan en función de tres variables: la opción, el número de segundos y el umbral del grado de similitud. La variable "Opción" indica el número de parámetros que el anunciante ha seleccionado como se muestra en la Tabla 3.5. Por ejemplo, en la opción 2 el anunciante selecciona el navegador, la ciudad, el país, la hora y la palabra clave. La variable "Segundos" indica el tiempo máximo para calcular una solución. La variable "Umbral" representa el mínimo grado de similitud que debe tener una visita con los requisitos del anunciante para que se pueda mostrar el anuncio.

Algoritmo 3.1 Hilos aplicados al intercambio de anuncios.

```
1:  Programa principal
2:  for Visita = 1 to 1,000 do
3:      for j = 1 to 100,000 do
4:          Crear-hilo(j)
5:      end for
6:      for j = 1 to 100,000 do
7:          Ejecutar-hilo(j)
8:      end for
9:      while (hilos-acaban) do
10:         Esperar()
11:     end while
12:     Guardar-mejor-solución()
13: end for
14: Fin Programa principal
```

Algoritmo 3.2 Pseudocódigo de la función Crear-hilo (int k).

```
1:  Función: Crear-hilo(int k)
2:  Leer-anunciantes-del-fichero(k)
3:  Comparar-anunciantes-con-visita()
4:  if (Solución > solución-global) then
5:      Solución-global ← solución
6:  end if
7:  if (Último-hilo()) then
8:      Escribir-solución-fichero(Solución-global);
9:  end if
10: Fin Función Iniciar hilo
```

3.5.1.2. Resultados de los hilos y de las matrices de similitud

Teniendo en cuenta que en el intercambio de anuncios pueden colaborar varios millones de anunciantes, es importante diseñar un algoritmo que encuentre entre todos ellos el más

Lenguaje	Ca	En	En-Gb	En-Us	Es	Es-419
Ca	1	X	X	X	X	X
En	0	1	X	X	X	X
En-Gb	0	0,9	1	X	X	X
En-Us	0	0,8	0,7	1	X	X
Es	0	0	0	0	1	X
Es-419	0	0	0	0	0,9	1

Tabla 3.10: Matriz de similitud para el parámetro: Idioma del Sistema Operativo.

adecuado para una visita. Este algoritmo debe ejecutarse en pocas décimas de segundo por lo que tendrá que utilizar varios hilos que se ejecuten a la vez.

En la Tabla 3.10, que es una versión reducida de la original, se observa que el grado de similitud entre los países que tienen el castellano como lengua oficial es muy elevado. Debido a que la página web está en castellano, la mayoría de las visitas tienen este idioma. En total se ha creado 12 tablas, una por cada parámetro, para poder aplicar las matrices de similitud.

Con un valor de 1 para el umbral y un valor de 2 para la opción, el algoritmo se tomó un total de 24 minutos, 33 segundos y 76 centésimas para comparar una única visita con 10.000.000 campañas de los anunciantes. Si se tiene en cuenta que el tiempo máximo establecido es de 0,1 segundos, entonces se deduce que este algoritmo es inservible.

Debido a que los tiempos son muy elevados, se ha establecido un número máximo de segundos a partir del cual ya no se procesan más hilos. Lógicamente, cuanto mayor sea el número de segundos más hilos podrán ejecutarse y por tanto, mejor será la solución. Cuanto menor sea el umbral más visitas cumplirán los requisitos del anunciante y mejores serán los resultados. Los valores que se muestran en la Tabla 3.11 representan el valor medio del $Ranking\ Anuncio$. Cuanto mayor sea este valor, mejor calidad tendrán los anuncios mostrados.

Si se observa la Tabla 3.12, el número de comparaciones aumenta si se utilizan matrices de similitud. Esto se debe a que hay que recorrer toda la matriz de similitud. En primer lugar, se recorren las filas comparándolas con el parámetro de la visita y posteriormente se recorren las columnas comparándolas con el valor de la campaña. Si los resultados obtenidos son 3 y 5, la celda [3,5] de la matriz contendrá el grado de similitud entre estos dos valores.

Para calcular el resultado se multiplica el número de hilos que está en la columna hilos con la matriz de similitud por el número de campañas que tiene cada hilo que son 100. Ese resultado se multiplica por el número de comparaciones que se realizan con y sin la matriz de similitud para cada opción.

En la columna “Hilos con matriz de similitud” se muestra el número de hilos que procesa

	Umbral	1 Seg	2 Seg	3 Seg	5 Seg	10 Seg	15 Seg	25 Seg
Opción 1	0,7	8,77	8,93	9	9,05	9,11	9,16	9,38
	0,8	8,78	8,93	9	9,05	9,1	9,16	9,4
	0,9	8,78	8,93	8,98	9,06	9,11	9,19	9,37
	1	8,8	8,94	8,99	9,06	9,11	9,17	9,37
Opción 2	0,7	8,45	8,62	8,72	8,8	8,88	9,01	9,17
	0,8	8,28	8,54	8,64	8,72	8,85	8,92	9,09
	0,9	6,28	6,89	7,1	7,35	7,65	7,81	7,98
	1	5,01	5,59	6,03	6,26	6,95	7,11	7,54
Opción 3	0,7	8,5	8,7	8,79	8,88	8,91	9,05	9,17
	0,8	7,5	7,92	8,11	8,29	8,49	8,57	8,7
	0,9	4,5	5,18	5,55	5,96	6,48	6,71	7,04
	1	0,08	0,19	0,25	0,38	0,51	0,74	1
Opción 4	0,7	8,1	8,35	8,49	8,64	8,75	8,83	9,01
	0,8	6,58	7,05	7,34	7,63	7,97	8,09	8,25
	0,9	3,58	4,16	4,64	4,93	5,66	5,97	6,29
	1	0,02	0,06	0,05	0,09	0,17	0,19	0,35

Tabla 3.11: Cobertura de los anuncios en función del umbral de la similitud.

Opc.	Comp. usando matriz similitud	Comp. sin usar mat. de simil.
1	107.600.000	107.400.000
2	1.212.300.000	117.273.810
3	2.035.700.000	134.354.215
4	2.434.900.000	144.191.923

Tabla 3.12: Número de comparaciones para 1.000.000 campañas.

el algoritmo en ese intervalo de tiempo. Como se ha comentado anteriormente, la mayor parte del tiempo se invierte en acceder a los ficheros por lo que el número de hilos es similar usando la matriz de similitud que sin usarla. Cabe aclarar que el resultado del número de hilos se realizó cuando se utilizó la matriz de similitud y puede ser más ilustrativo mostrar el número de comparaciones sin usar la matriz de similitud en función del mismo número de hilos.

3.5.2. Árboles AVL para optimizar el intercambio de anuncios

El árbol AVL toma su nombre de las iniciales de los apellidos de sus inventores: Adelson, Velskii y Landis [168]. Es un árbol binario de búsqueda que cumple con la condición de que siempre están equilibrados, de tal modo que para todos los nodos, la altura de la rama izquierda no difiere en más de una unidad de la altura de la rama derecha o viceversa. Un árbol binario de búsqueda es una estructura de datos que permite organizar la información en función de algún atributo.

Cada nodo del árbol debe cumplir con la siguiente característica: los nodos inferiores a la izquierda de dicho nodo deben contener valores menores, los nodos inferiores a la derecha deben contener valores mayores.

3.5.2.1. Desarrollo de algoritmos usando árboles AVL

Para mejorar los costos computacionales de este algoritmo se han empleado los árboles AVL y la codificación *hash*. Lo que se ha hecho es codificar con un *hash* los valores de cada una del millón de campañas de los anunciantes y añadirlo como un nodo a un árbol AVL con un atributo para el valor del $Ranking\ Anuncio$.

Este algoritmo codifica mediante un *hash* los valores de los campos de un anunciante y posteriormente los añade como nodos al árbol, cada nodo tiene una clave formada por un *hash* de una cadena de texto que representa esa combinación de parámetros y un valor que representa el valor del $Ranking\ Anuncio$.

Por ejemplo, si el anunciante decide configurar los siguientes parámetros con los siguientes valores: Hora=21, Navegador= Firefox, Versión navegador = 14.0.1, Sistema Operativo = Windows, País = Spain y Ciudad = Pamplona.

En ese caso, el valor de la cadena es:

“21Firefox14.0.1WindowsXPSpainPamplona”.

Cuando se aplica la función *hash* a esa cadena el valor es:

“2C1ECBEA35C21B712410CE7F7D0BB”.

Opc.	Seg.	Prom. Ranking	Prom. Comp.
1	1,36	9,89	16,65
2	1,55	8,45	30,42
3	1,84	1,24	49,74
4	1,92	0,44	51,03

Tabla 3.13: Resultados obtenidos con el algoritmo AVL.

3.5.2.2. Resultados obtenidos con los árboles AVL

El tiempo necesario para procesar 100.000 visitas con 1.000.000 anunciantes con este algoritmo es 1,66 segundos, lo que significa que el algoritmo se ejecuta aproximadamente 9.2 millones (9.206.250) veces más rápido que con los hilos.

Esto es debido a que el algoritmo no accede a los archivos, pues el árbol puede cargarse en la memoria, y se ha reducido el número de comparaciones por visita para la opción 2 a 1.172.738.107 comparaciones, con la opción "Usando la matriz de similitud" en sólo 51,03 comparaciones con árboles AVL como se aprecia en la Tabla 3.13.

El número medio de comparaciones que se realizan es de 16,65 para la opción 1, para ello se ha calculado la media de 100.000 visitas y los resultados son óptimos pues se han comparado todos y cada uno de los 1.000.000 de anunciantes.

3.5.3. Utilizando árboles de varios nodos

Para mejorar el rendimiento de los anuncios se ha probado utilizar árboles con múltiples nodos. En el primer nivel estarán todos y cada uno de los posibles valores que pueden tener los anunciantes.

3.5.3.1. Algoritmos utilizando árboles de múltiples nodos

Cada nodo del primer nivel tiene un número de hijos que representa los posibles valores que puede tener el segundo parámetro si el primer valor del parámetro coincide con el valor de ese nodo. Por lo tanto, si el primer parámetro tiene 29 valores diferentes el primer nivel del árbol tendrá 29 hijos. Si el hijo número 7 tiene como segundo parámetro 12 posibles valores, este nodo tendrá 12 hijos. Y lo mismo sucede con el resto de los niveles. El último valor contendrá el valor del $Ranking\ Anuncio$.

Para resolver el algoritmo se han probado tres opciones que son:

1. Árboles sin ordenar: Con esta opción se forma el árbol a partir de los requisitos de los anunciantes. Los posibles valores se van añadiendo al árbol según se van procesando las campañas.

		No ord.		Ord. + bús.bin.		Ord. por frec. + bús.bin.	
Opc.	**Result.**	**Seg.**	**Prom. Comp.**	**Seg.**	**Prom. Comp.**	**Seg.**	**Prom. comp**
1	9,89	0,58	20,59	0,78	37,95	0,53	18,52
2	8,45	1,18	50,9	1,17	75,68	1,03	43,59
3	1,24	1,81	73,25	1,86	99,75	1,69	64,93
4	0,44	1,61	74,98	1,65	102,42	1,65	66,64

Tabla 3.14: Resultados de árboles múltiples nodos de 100.000 visitas.

2. Árboles ordenados: En este apartado se hace exactamente lo mismo que en el primero pero posteriormente se ordenan los valores alfabéticamente según el nombre del parámetro. Esto se hace para utilizar la búsqueda binaria descendiendo desde la raíz del árbol hasta las hojas para obtener el valor $Ranking\ Anuncio$.

3. Árboles ordenados según frecuencia: En esta opción se hace lo mismo que en la primera pero se ordenan los árboles según la frecuencia con que los anunciantes demandan un parámetro. Si la mayoría configura como hora las 13:00 entonces será el hijo más a la izquierda y por el que se hará una primera comparación.

3.5.3.2. Resultados obtenidos con los árboles multinodo

Como se puede observar por el número de comparaciones: la mejor opción es ordenarlos por frecuencia, la segunda mejor opción es ponerlos sin ordenar y en tercer lugar aplicar la búsqueda binaria. Muchos de los nodos del árbol pueden estar formados por cuatro o cinco nodos para los cuales realizar una búsqueda binaria no tiene mucho sentido.

También se descarta la opción de ordenarlos por frecuencia pues el algoritmo tiene de media un número de comparaciones 0,07 % inferior como se aprecia en la Tabla 3.13, lo cual no justifica tener que estar ordenando el árbol cada vez que se añade una campaña.

3.5.4. Intercambio de anuncios mediante Apache Hadoop

Una de las formas más sencillas de resolver el problema del intercambio de anuncios y que supone menos quebraderos de cabeza, consiste en utilizar la famosa tecnología desarrollada por Apache y conocida como Hadoop[3]. Este lenguaje está orientado a aprovechar los *clusters* que se forman con miles de computadoras para formar las supercomputadoras.

Las grandes empresas como Yahoo, Amazon o Google, cuentan con este tipo de estructuras puesto que necesitan ejecutar algoritmos que procesan millones de datos. Esta plataforma tiene un lenguaje de programación llamado Hive y otro llamado Pig Latin que es el

[3]Apache Hadoop es un *framework* que está orientado a dar soluciones a los problemas relacionados con el Big Data como es el caso que se plantea y que permite solucionar este problema con unas pocas líneas.

que se ha utilizado para resolver este problema. El Algoritmo 3.3 no es tan eficiente como el de los árboles, puesto que tiene un costo computacional 60,5 veces superior a los árboles AVL y 80,5 a los árboles con múltiples nodos. Sin embargo, permite resolver el problema con tan solo diez líneas de código.

Algoritmo 3.3 Código Apache Hadoop para el intercambio de anuncios.

```
1:                                                          ▷ ANUNCIANTES
2:                                                ▷ Cargar la tabla de los anunciantes
3:  Anun0 = Load 'default.anunciantes2'
4:  USING org.apache.hcatalog.pig.HCatLoader();
5:                                              ▷ Seleccionar las columnas que interesan
6:  Anun1 = Foreach Anun0 Generate $2, $4, $8, $9, $10, $11, $12, $15,$19*$20;
7:                                                          ▷ Agrupar las filas
8:  Anun2 = Group Anun1 by ($0,$1,$2,$3,$4,$5,$6,$7);
9:                                           ▷ Seleccionar el máximo de cada grupo
10: Anun3 = Foreach Anun2 Generate group, MAX(Anun1.$8);$
11:                                            ▷ Se convierte cada tupla de cada fila
12: Anun4 = Foreach Anun3 Generate FLATTEN($0),$1;
13:                                        ▷ Cargar la tabla de anunciantes en memoria
14: Visitas0 = Load 'default.visitas'
15: USING org.apache.hcatalog.pig.HCatLoader();
16:                                 ▷ Por cada celda seleccionar las columnas que interesan
17: Visitas1 = Foreach Visitas0 Generate $2, $4, $8, $9, $10, $11, $12, $15;
18:                                          ▷ Crear una tabla en la que coincidan los
19:                                        ▷ Campos de las visitas y de los anunciantes
20: Visitas2 = Join Visitas1 by ($0,$1,$2,$3,$4,$5,$6,$7),
21: Anun4 by ($0,$1,$2,$3,$4,$5,$6,$7);
22:                                                      ▷ Seleccionar columnas
23: Res = foreach Visitas2 generate $0,$1,$2,$3,$4,$5,$6,$7,$16;
24:                                                       ▷ Guardar al respuesta
25: store Res into 'Respuestas4';
26: DUMP Res;
```

En la Tabla 3.15 se muestran los resultados obtenidos y también los tiempos necesarios para llegar a esos resultados. Las pruebas se han realizado con 100.000 visitas y con 1.000.000 de campañas publicitarias.

3.6. Conclusiones

En primer lugar, se puede concluir que el intercambio de anuncios es beneficioso para las redes publicitarias tanto para mejorar el rendimiento de sus campañas publicitarias como para la detección de fraude. Posteriormente, se ha realizado un experimento en el que se ve

Opc.	Seg.	Result.
1	214	9,89
2	226	8,6
3	260	1,24
4	309	0,44

Tabla 3.15: Resultados aplicando Apache Hadoop.

como aumenta la cobertura de los anuncios conforme más redes colaboran. Y como disminuye cuanto más específicas sean las campañas. Por último, para el intercambio de anuncios entre redes se han desarrollado algoritmos basados en hilos, en árboles y en Hadoop.

Viendo los resultados obtenidos en los experimentos, concluimos que la opción de utilizar "Hilos" no es apropiada por la enorme cantidad de tiempo que se necesita para procesar el algoritmo. Uno de los motivos por los que el tiempo fue tan alto se debe a que a la plataforma le cuesta bastante tiempo tener que abrir 100 mil ficheros y por este motivo se incrementó el tiempo de ejecución.

Los árboles AVL mediante *hash* son los que mejores resultados han proporcionado. Pero tienen dos desventajas, en primer lugar hay que crear el árbol ordenado por cada conjunto de parámetros que se quiera comparar, lo cual ocupa un gran espacio en memoria. En segundo lugar no permitiría utilizar la matriz de similitud mediante el umbral de parecido. Para poder implementar este tipo de lógica habría que codificar en *hash* todas las relaciones que tiene un parámetro con otro y se tendrían que añadir todas las posibles combinaciones haciéndolo casi inviable, pues aumentaría de forma exponencial el número de nodos del árbol.

Otra desventaja que se podría añadir es el costo computacional de la creación del árbol pero esto no es muy preocupante porque se puede realizar mientras el programa se está ejecutando. Es decir, no se crea en el instante en que el usuario hace la visita y por lo tanto no es un costo computacional crítico.

Los árboles de varios nodos parecen la opción más acertada pues el tiempo de ejecución es relativamente bajo y tienen la ventaja que permiten el uso de matrices de similitud, a diferencia de los árboles binarios. Esto se podría realizar simplemente con un algoritmo de *backtracking* que vaya recorriendo el árbol y que pode una ruta si se ha superado el umbral de similitud. Para este problema en particular, ordenarlos no supone una gran ventaja, pues según los resultados obtenidos no hay una gran diferencia en el número de comparaciones.

Por último, teniendo en cuenta la capacidad que tienen algunos equipos de hardware, utilizar Pig Latin de Hadoop es una buena solución cuando el algoritmo se tiene que ejecutar pocas veces, o cuando el tiempo de ejecución no sea algo prioritario. Pues el código que hace falta para desarrollar el algoritmo se puede resumir en escasas líneas lo que hace que haya muy pocos errores. Además la matriz de similitud se puede implementar mediante funciones

hdfs que puede implementar el usuario en lenguajes como Python o Java.

Como posibles líneas de trabajo futuro, se podría ofrecer a los anunciantes hacer campañas optimizadas informando al anunciante sobre los parámetros que permitirán obtener mayores ingresos en las campañas de publicidad. No haciendo necesario contratar a un experto que evalúe y analice los resultados.

Parte II

Diseño de un modelo multicriterio de intercambio de anuncios entre pequeñas redes para garantizar la satisfacción de todos los roles

Capítulo 4

Diseño de un modelo multicriterio de intercambio de anuncios y su optimización mediante un algoritmo genético

4.1. Introducción

Las campañas de publicidad en Internet se han ido orientando poco a poco a nichos más específicos. Las pequeñas redes publicitarias han ido desapareciendo paulatinamente pues han sido incapaces de ofrecer a los anunciantes campañas dirigidas a un pequeño segmento debido a su escaso número de visitantes.

Para solucionar este problema se han creado enfoques como el *Real-Time Bidding* (RTB), que se puede ver como una gran subasta donde millones de anunciantes pujan por mostrar su anuncio en un espacio publicitario.

Para ello, se han creado metodologías basadas en fomentar y aplicar las técnicas de aprendizaje automático conocidas como subastas *online* [169]. Estos modelos pueden predecir de manera precisa el grado de aceptación de un usuario para un anuncio, por lo que la probabilidad de compra aumenta de forma considerable [12]. Otros autores sugieren que para optimizar el rendimiento en la selección de anuncios debe tenerse en cuenta no solamente la parte semántica entre las palabras clave y las consultas, sino también otros factores como el CTR histórico o la satisfacción de los usuarios al visitar la página.

Otros autores se centran en optimizar la satisfacción de los anunciantes, partiendo de la premisa de que los anunciantes estarán más dispuestos a hacer inversiones si obtienen un buen rendimiento. En este sentido, hay que destacar las investigaciones de Balseiro, que hace

un análisis profundo sobre el equilibrio que debe existir entre el rendimiento económico, la selección del anuncio más rentable y la calidad que se ofrece a los anunciantes [6].

En este capítulo se desarrolla un Modelo de Intercambio de Anuncios (MIA) entre las pequeñas redes de publicidad para que puedan competir con las grandes redes. El MIA busca habilitar a las pequeñas redes para ofrecer a los anunciantes campañas orientadas con suficientes impresiones con el fin de competir contra las grandes redes.

Para ello, se desarrolla los pasos necesarios para diseñar un MIA. Los principales pasos son definir los objetivos para realizar un MIA de forma exitosa, definir un conjunto de medidas orientadas a luchar contra el fraude, definir una función para evaluar el rendimiento del modelo en función de los objetivos establecidos y desarrollar una metodología basada en un algoritmo genético para calcular los pesos óptimos en la función de selección de un anuncio.

El MIA debe garantizar que todas las partes involucradas (anunciantes, editores y especialmente las redes publicitarias) obtengan ganancias considerables. El rendimiento obtenido por cada una de las redes que participa en el MIA debe ser superior al rendimiento que genera cada red que opera de forma independiente. En el proceso de colaboración, el MIA intercambia información sobre los requisitos de los anunciantes, características de los sitios web de los editores, y sobre los perfiles de los usuarios.

El objetivo de este capítulo se puede resumir en una sola idea: Calcular los pesos óptimos asociados a cada objetivo en la fórmula de selección de un anuncio que optimicen el rendimiento del modelo en base a una métrica expresada en términos económicos. Esta investigación es de gran interés, pues la publicidad *online* ha experimentado un gran crecimiento pero se han publicado muy pocos estudios de investigación científica frente a los problemas de las pequeñas redes y sus posibles soluciones.

En primer lugar, se describen brevemente algunos estudios sobre los modelos de intercambio de anuncios y las técnicas de optimización mediante algoritmos genéticos. Posteriormente se explica el MIA en términos generales. Se ilustra la estructura y cada uno de los módulos que componen el MIA, especialmente el módulo de "Selección del anuncio". Posteriormente, se enumeran los principales objetivos para un correcto funcionamiento del MIA, las reglas para evitar el fraude en línea y las penalizaciones para asegurar el cumplimiento de los objetivos comunes. También se describe una metodología para optimizar la función de selección de anuncios.

A continuación se describen los experimentos desarrollados y se redacta un breve análisis de los resultados obtenidos. Para concluir, se presentan las conclusiones de esta investigación y algunas posibles líneas de investigación para futuros trabajos.

4.2. Diseño del modelo de intercambio de anuncios

A través de un MIA las redes pueden intercambiar anuncios entre sí para ser más competitivas. Trabajando de manera independientemente las visitas de aquellos usuarios que no corresponden a alguno de los requisitos de cada anunciante se desperdiciarían. Pero si las redes cooperaran entre sí, los demás anunciantes podrían aprovechar estas visitas.

La colaboración entre redes no solo mejora el desempeño económico, sino que también mejora la detección de fraude, pues las redes podrían compartir información acerca de las técnicas fraudulentas de los anunciantes y de los editores[1]. También podrían compartir información sobre otras amenazas como son los *click-bots* y las *click-farms* [171]. La detección de fraude mejora la rentabilidad de las campañas pues los anunciantes no pagan por clics fraudulentos por los que los anunciantes no obtienen ningún beneficio.

Aunque la prevención de fraude es muy importante en la publicidad *online*, esta investigación se centra en el diseño de un MIA entre pequeñas redes como se muestra en la Figura 4.1. Cada red tiene sus propios editores y sus propios anunciantes.

Cuantos más editores tenga una red, mayores serán los ingresos que recibe, pues ellos son los propietarios de las páginas en las que aparecen los anuncios. Tener muchas visitas permite a los anunciantes lanzar campañas mejor orientadas y, por lo tanto, obtener un mayor rendimiento económico. Además, cuantos más anunciantes haya más aumentará el precio por clic, pues los anunciantes compiten entre ellos por ciertas palabras clave. Por último, cuantas más redes participen más aumentarán los beneficios, ya que cada red trae consigo más editores y más anunciantes.

Como se muestra en la Figura 4.1, en el modelo que se ha propuesto el intercambio de anuncios se lleva a cabo a través del Sistema de Intercambio de Anuncios (SIA). El MIA se compone básicamente del SIA y de todas las redes que participan en el intercambio de anuncios. Si el número de redes se vuelve muy grande se podría considerar replicar el SIA para asegurar que no aumenta el tiempo de respuesta y así equilibrar la demanda de anuncios. Sin embargo, esta investigación se limita a un único sistema.

El SIA es la piedra angular de este modelo ya que realiza todas las funciones necesarias para un intercambio de anuncios adecuado. Los procesos más importantes del SIA son seleccionar el mejor anuncio para que se muestre entre todos los candidatos [172], mantener el sistema de detección de fraude actualizado [33], y manejar la gestión de cobros y pagos.

[1] Respetar la privacidad de las personas en la información que manejan las empresas es algo imprescindible en el funcionamiento del negocio [4, 170].

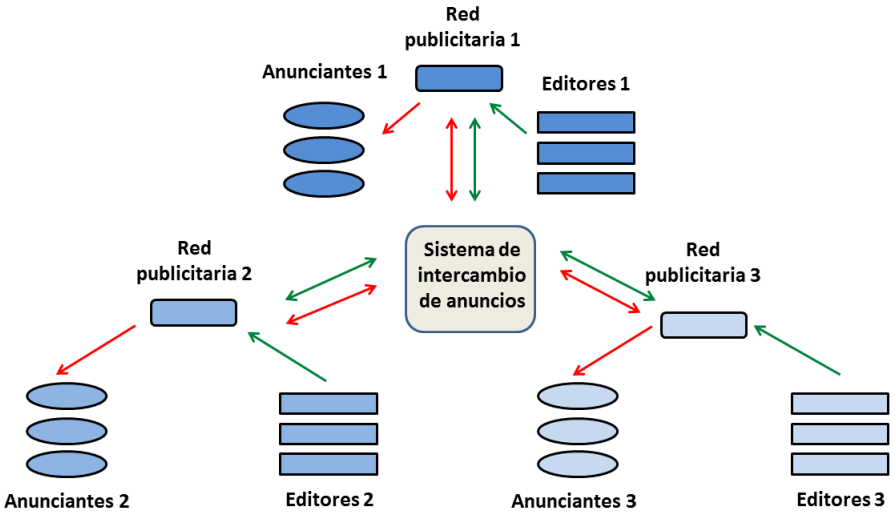

Figura 4.1: Publicidad de intercambios de la estructura del módulo.

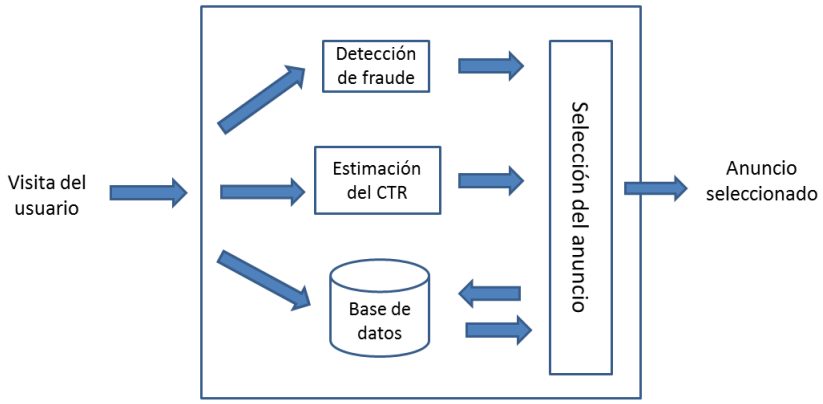

Figura 4.2: Estructura del sistema de intercambio de anuncios.

4.3. Sistema de intercambio de anuncios

Para desarrollar el MIA se propone el Sistema de Intercambio de Anuncios (SIA) como se muestra en la Figura 4.2. El SIA que se ha diseñado sólo admite el modelo de pago CPC, que es el más extendido, y se compone de cuatro módulos interconectados e independientes: el módulo de estimación de CTR, el módulo de detección de fraude, el módulo de selección de anuncio y la base de datos. Cada módulo está diseñado para un propósito diferente y todos ellos son necesarios para hacer posible el intercambio de anuncios.

El módulo más importante es el de selección de anuncios. Los otros tres módulos recogen la información necesaria para que el módulo “Selección de anuncios” elija el anuncio que proporcione mayor rendimiento. El módulo de estimación del CTR proporciona infor-

mación sobre la probabilidad de que un usuario haga clic en un anuncio. El módulo de detección de fraude informa sobre la probabilidad de que un anuncio sea de tipo *spam* y la posibilidad de que el clic sea fraudulento. Y la base de datos registra toda la información necesaria para llevar a cabo todos los procesos que intervienen en la publicidad *online*.

4.3.1. Módulo 1: Estimación de CTR

Teniendo en cuenta que sólo se usa el método de pago CPC en este modelo, las redes deben dar prioridad a los anuncios más rentables con el fin de maximizar sus ingresos.

El factor más importante para determinar la rentabilidad de un anuncio es el CTR. El CTR de un anuncio se calcula como el cociente entre el número de clics y el número de impresiones de dicho anuncio. Pero en el caso de una sola visita, el CTR puede ser expresado como la probabilidad de que un usuario genere un clic en el anuncio que aparece en una página web y esta probabilidad se expresa como un número real en el rango [0,1]. Estimar con precisión el CTR de un anuncio es uno de los mayores retos de la publicidad *online* [99].

Los métodos *machine learning* se han aplicado con éxito en la predicción del CTR [32]. Para ello, se entrena a los modelos con un conjunto de muestras con características de los usuarios y de las páginas web que visitan. Algunas características analizadas son el tamaño del anuncio, la posición del anuncio en la página o la categoría del anuncio.

Cuando el usuario genera un clic, la salida del modelo es "1" y cuando el usuario no genera un clic la salida es "0". Hay que aclarar que en lugar de predecir la clase del modelo, se predice la probabilidad de que el modelo pertenezca a una clase en el rango [0,1]. El CTR está representado por la ecuación 4.1.

$$P(Va|C)\, donde\, C = \{Clic,\ No\, clic\} \tag{4.1}$$

La mayoría de las redes CPC muestran el anuncio más valioso. El valor de un anuncio se calcula teniendo en cuenta la probabilidad de que se genere un clic multiplicado por el precio del clic, tal y como se expresa en la ecuación 4.2:

$$Valor\ Anuncio = P(CTR|Clic) \times Precio\, CPC \tag{4.2}$$

4.3.2. Módulo 2: Detección de fraude

El módulo de detección de fraude está diseñado para medir la probabilidad de que un anuncio sea de tipo *spam* y para medir la probabilidad de que un clic generado en la página web de un editor sea fraudulento.

La probabilidad de fraude en ambos casos se puede expresar mediante un número real entre cero y uno, $r \in R \in [0, 1]$. Como se ha mencionado anteriormente, calcular la probabilidad de fraude es un proceso altamente complejo. Por lo que resulta muy difícil determinar cuándo una persona está cometiendo fraude evaluando únicamente un clic o una impresión de anuncio. Para determinar si un anunciante o un editor está haciendo fraude es necesario evaluar un conjunto suficientemente grande de clics o de impresiones de anuncios[2].

Por otra parte, los modelos que determinan la probabilidad de fraude tendrían que tener en cuenta tanto el historial de los clics de los editores como el historial de las impresiones de los anunciantes. En el caso de los anuncios *spam*, se debe tener en cuenta información acerca de los anunciantes como la duración de la campaña del anunciante, el tipo de producto que se anuncia, el CTR del anuncio o el comportamiento de los usuarios cuando se muestra el anuncio [174].

En el caso de fraude por clic se deben revisar las características de los editores y también los datos sobre aquellos usuarios que visitaron la página [175]. Los factores más importantes implicados en la detección de fraude por clic son la distribución de las IPs, las horas de acceso más frecuentes, los países y ciudades con más visitas a la página, el tipo de usuarios que acceden[3] y el comportamiento de los usuarios antes y después de generar un clic[4].

Para decidir si se expulsa a un editor o a un anunciante del MIA se requiere revisar un número suficiente de clics. Para esta investigación se simplifica el problema utilizando una función aleatoria que simule el comportamiento de los anunciantes y de los editores, entre los cuales habrá algunos que cometan fraude.

La probabilidad P de que un clic x_i sea fraudulento se puede expresar como $P(x_i|fraude) = \alpha$, y $P(x_i|\in fraude) = 1 - P(x_i|fraude)$.

4.3.3. Módulo 3: La base de datos

La base de datos almacena todos los datos requeridos para optimizar el rendimiento del MIA sobre los anunciantes, los editores y las redes. Los datos más importantes almacenados en la base de datos contienen información relacionada con los pagos de los anunciantes y los gastos de los editores. Además, se almacena información sobre los fraudes cometidos y la información utilizada por la función de selección de un anuncio como son el CTR del anuncio o el CPC de los anunciantes.

[2]Este es el caso de Google, que no da un pago a los editores hasta que alcanzan un número suficiente de clics que permita determinar con precisión si el editor es fraudulento [173].

[3]El acceso de los usuarios puede ser directo, a través de un enlace en otra página o a través de un motor de búsqueda [176].

[4]El comportamiento de los usuarios consiste en analizar varios datos como el tiempo que estuvieron en la página, los movimientos del ratón, o las secciones de la página que consultaron [42].

Cuando un usuario realiza una visita a una página, se muestra un anuncio y se actualiza la base de datos. La probabilidad de que un clic sea fraudulento y de que el anuncio sea *spam* también se actualiza.

Existen otras variables, tales como el nivel de satisfacción de la red y los resultados de la campaña que pueden ser actualizados cada N visitas. Donde N es un entero que si es demasiado grande, el algoritmo no será eficaz puesto que las variables no estarán actualizadas y si N es demasiado pequeño, el proceso será computacionalmente muy costoso. Por lo tanto, se define $N = 1{,}000$.

4.3.4. Módulo 4: Selección del anuncio

El módulo más importante es el de "Selección de un anuncio", ya que es responsable de seleccionar el anuncio que se muestra al usuario que accede a la página de un editor de entre todos los candidatos. Este módulo tiene que cumplir con varios objetivos del MIA como son maximizar los ingresos de los editores, equilibrar los anuncios entregados y los recibidos para cada red, y asegurar la satisfacción de los anunciantes.

Cada vez que un usuario accede a la página web de un editor se realiza una selección de entre todos los anuncios y se descartan los anuncios que pertenecen a una categoría diferente al de la página web del editor que se accede. Los anuncios que se descartan son llamados candidatos. Seguidamente, se selecciona un único anuncio entre todos los candidatos, es decir, aquel que posee el máximo valor del $Ranking\ anuncio$. Para seleccionar el mejor anuncio se aplica la función $F(Anuncio)$, que asigna un valor real en el intervalo [0,1] a todos los anuncios candidatos. Este valor se llama $Ranking\, anuncio$ y se explica en detalle en la subsección 4.3.5.4.

La función $F(Anuncio)$ incluye pesos que son asignados en proporción a la importancia de cada objetivo. Para calcular el valor óptimo de esas ponderaciones se aplicá un algoritmo genético. El valor $Ranking\ anuncio$ se calcula teniendo en cuenta todos los objetivos del MIA. Las grandes compañías publicitarias han desarrollado su propia metodología para calcular el $Ranking\, anuncio$. Sin embargo, dichos algoritmos nunca han sido revelados, ya que representan su ventaja competitiva sobre el resto de plataformas [177].

Como se puede observar en la Figura 4.2, este módulo toma en cuenta el CTR y la probabilidad de que los anunciantes y editores comentan fraude. También consulta y actualiza la base de datos donde se ofrece información sobre las campañas de los anunciantes, el estado de cuenta de los editores y el rendimiento de las redes del MIA.

4.3.5. Desarrollo del modelo de intercambio de publicidad

Para desarrollar el MIA se siguen varios pasos: En primer lugar, se definen los objetivos necesarios para garantizar el funcionamiento adecuado del ecosistema publicitario. Para garantizar que se cumplan los objetivos se define una penalización económica por cada objetivo, de tal manera que cuanto menos se cumplan los objetivos mayores serán las penalizaciones.

Además, se creará un conjunto de reglas para evitar las actividades fraudulentas en el MIA. También se ha establecido una métrica expresada en términos económicos con el fin de medir el rendimiento del MIA. Seguidamente se muestra el algoritmo en pseudocódigo que utiliza el MIA y se define una metodología para encontrar la configuración óptima de los pesos mediante un algoritmo genético para la función de selección de anuncios.

4.3.5.1. Definición de los objetivos para el MIA

Para tener éxito en el MIA se deben cumplir varios objetivos [178]. Algunos de los objetivos pueden tener intereses enfrentados. Por ejemplo, el MIA debe generar unos ingresos tan altos como sea posible de tal manera que los editores pueden ser mejor pagados. Pero, al mismo tiempo, el MIA no debe cobrar a los anunciantes un precio tan alto que haga que sus campañas no sean rentables. Por lo tanto, debe haber un equilibrio en el precio que satisfaga a los anunciantes, a los editores y a las redes.

Además, hay muchos asuntos que el MIA debe de resolver como por ejemplo los anuncios de tipo *spam* o los editores fraudulentos. Por lo tanto, se deben diseñar un conjunto de reglas para defender el MIA frente a estas amenazas.

Los objetivos del algoritmo para cada anuncio a_i perteneciente a los anunciantes $a_i \in A$ y a los editores $p_i \in P$ de la publicidad de la red publicitaria RP_i, donde una $RP_i \in MIA$, son:

- **(O1) Impresiones de anuncios por anunciante:** Todos los anunciantes necesitan mostrar una cantidad razonable de anuncios para que estén satisfechos. Si el algoritmo se centra en maximizar los ingresos del MIA, algunos anunciantes pueden quedarse sin impresiones. Por ejemplo, se podría considerar el caso en que dos anunciantes seleccionen la misma categoría. Si el primer anunciante establece un precio de $0,62 y el segundo establece un precio de $0,55, el segundo no tendrá impresiones. Por lo tanto, se debe garantizar una distribución equitativa del número de impresiones donde los anunciantes que paguen un precio más alto tengan la ventaja de que sus anuncios se muestran con mayor frecuencia.

- **(O2) Anunciantes *spam*:** Muchos anunciantes muestran anuncios en Internet con una

intención maliciosa. Estos anuncios son conocidos en la publicidad *online* como *spam*. Los anuncios *spam* tienen como objetivo obtener beneficios por engañar a los usuarios.

Los anunciantes *spam* son muy perjudiciales para el ecosistema publicitario por lo que se debe calcular la probabilidad de que un anuncio sea *spam* para reducir tanto como sea posible las veces en las que se muestran. También se debería tener un equipo encargado de verificar si un anunciante está tratando de engañar a los usuarios cuando el sistema alerta de que un anunciante puede estar haciendo trampa.

- **(O3) Rentabilidad de campañas:** Algunos anunciantes inexpertos podrían pagar por sus campañas un precio por encima del precio de mercado. No es recomendable sacar provecho de este tipo de anunciantes y cobrarles un precio más alto. El MIA debe hacer campañas rentables para todo tipo de anunciantes. Por lo tanto, se debe asegurar que en el MIA P_a es similar a P_m, $P_a \backsimeq P_m$, donde P_a es el precio pagado por un anunciante $a_i \in A$ y P_m representa el precio de mercado.

- **(O4) Equilibrio de la redes de publicidad:** Mediante la colaboración, todas las redes deberían poder usar el espacio de otras redes para mostrar anuncios. Si se quiere que todas las redes participen en el MIA, entonces el número de anuncios A_r recibido por cada red será similar al número de anuncios entregado A_d, es decir $A_r - A_d \backsimeq 0$.

- **(O5) Fraude por clic de los editores:** El fraude cometido por los editores es conocido como fraude por clic y puede llegar a ser muy perjudicial para las campañas publicitarias. Este tipo de fraude se realiza con la intención de aumentar los ingresos de los editores o de dañar la plataforma *online*[5]. Debido al fraude por clic, los anunciantes terminan pagando por clics que no les proporcionan ningún beneficio. Esto aumenta la probabilidad de que los anunciantes se vayan a redes que ofrezcan campañas más rentables.

- **(O6) Maximización de ingresos**: Aunque este es el objetivo más importante, se ha puesto en última posición porque cada uno de los objetivos anteriores excepto este tiene una penalización asociada. El algoritmo de selección de anuncios debe buscar los anuncios más rentables con el fin de distribuir la mayor cantidad de ingresos posibles. Los editores deben obtener unos ingresos económicos razonables para no abandonar la red publicitaria y además consigan reclutar nuevos anunciantes. Unos ingresos más altos permitirían pagar mejor a los editores.

[5]Muchos editores pueden hacer clic en sus anuncios o decirles a sus amigos que lo hagan. También existen *click-bots* cuyo objetivo es generar clics para dañar el ecosistema publicitario.

4.3.5.2. Penalizaciones económicas para el MIA

Para garantizar que se cumplen todos los objetivos se define una sanción económica P_i y un coeficiente X_i asociado con cada penalización. De tal manera que cada penalización se aplicará siempre que su correspondiente objetivo no se cumpla[6]. Los coeficientes X_i permiten aumentar o disminuir la penalización económica que se aplica cuando un objetivo no se cumple.

Calcular los coeficientes exactos X_i para cada penalización no es tarea fácil y se deben calcular basándose en datos históricos y análisis estadísticos. La metodología para calcular el valor de estos coeficientes no es uno de los objetivos de esta investigación.

Las cinco penalizaciones que se han definido son:

- **(P1) Porcentaje de anuncio de impresión:** Se debe aplicar una penalización por cada anunciante que no logró publicar un número suficiente de anuncios. Para averiguar el número de impresiones que corresponde a cada anunciante se calcula el número de visitas de cada categoría y se multiplica por el CPC del anuncio. Por lo tanto, P1 puede ser expresado como: *"Por cada anunciante cuyo promedio de impresiones de publicidad se encuentre por debajo del 25 %, se le restará a los ingresos X_1 veces los ingresos promedio de los anunciantes de la red"*.

- **(P2) Anunciantes de *spam*:** Los anuncios de tipo *spam* tienen un impacto negativo en las redes. La publicación de este tipo de anuncios se acaba traduciendo en pérdidas económicas pues los usuarios y los editores terminan perdiendo confianza en la plataforma. Por lo tanto, es necesario tomar alguna acción para evitar que se publiquen estos anuncios. Para ello, se define P2 como: *"Por cada clic de un anunciante de tipo spam se resta X_2 veces el dinero generado a los ingresos totales"*.

- **(P3) Rentabilidad de las campañas:** La rentabilidad de las campañas tiene una gran importancia para garantizar que los anunciantes estén satisfechos con la plataforma. Por ello, se pretende evitar cualquier abuso sobre los anunciantes inexpertos, los cuales pueden pagar un precio por encima del precio de mercado. Para evitar este tipo de abusos se define una penalización que se traduce como un valor económico negativo cada vez que las campañas de los anunciantes no sean rentables. P3 se puede expresar como: *"Por cada anunciante que paga un precio de un 25 % por encima del precio de mercado, se resta a los ingresos totales X_3 veces el dinero generado por ese anunciante"*.

[6]La lógica de estas penalizaciones es que los participantes (redes, anunciantes y editores) que no estén satisfechos con el MIA generalmente dejan la plataforma, lo que se traduce en pérdidas económicas.

- **(P4) Balance de las redes de publicidad:** Cuando una red no está equilibrada es posible que deje de trabajar con la plataforma. Por ello, se utiliza una variable para expresar lo equilibrada que está una red. Dicha variable determina la relación entre el número de anuncios entregados y los recibidos. Sólo se castiga a las redes que entreguen más anuncios de los que reciban. Por lo tanto, la penalización P4 dice: *"Por cada red que reciba el 25 % menos de anuncios del número de anuncios que ofrece, se reducirá X_4 veces los ingresos de dicha red a los ingresos totales".*

- **(P5) Fraude por clic de los editores:** Como se mencionó anteriormente, el fraude por clic hace que las campañas de los anunciantes fracasen y por lo tanto, hace que haya más probabilidades de que los anunciantes pasen a otras plataformas publicitarias. Para evitar esto se define la penalización P5, que dice: *"Por cada clic fraudulento de un editor se restará X_5 veces el valor de este clic al total de ingresos".*

4.3.5.3. Políticas en el MIA y reglas contra el fraude

Es posible engañar a los anunciantes con la intención de obtener una gran cantidad de dinero en el corto plazo y posteriormente retirarse del negocio. Cabe destacar que en la actual investigación, el fraude constituye no solamente un asunto económico sino también una cuestión de ética. El objetivo es hacer el MIA lo más eficiente en el largo plazo.

Se asume que al transmitir transparencia y confianza a todos los anunciantes, se mejore el rendimiento del MIA. No se puede perder de vista el hecho de que los anunciantes son lo más importante en cualquier MIA, puesto que son los que hacen posible el negocio. Por lo tanto, se debe definir un conjunto de políticas y de normas orientados a defender sus intereses.

- **Políticas en el MIA:** Se debe tener en cuenta que muchos editores expulsados de otras plataformas de publicidad por cometer fraude pueden llegar a la plataforma y aplicar las mismas técnicas fraudulentas. Además, se ha de establecer una política muy estricta con respecto al fraude, pues deben diseñarse sistemas muy eficientes. La detección de fraude es algo muy complejo que constituye una barrera de entrada a este negocio debido a la alta inversión requerida.

  Cualquier editor que quiera participar en el negocio deberá aceptar las políticas del MIA que están encaminadas a reducir el fraude lo máximo posible. Estas políticas buscan expulsar a los editores antes de que reciban cualquier pago si el grupo de expertos del sistema determina que el fraude ha sido cometido de manera intencionada[7].

[7]Por ejemplo, Google es muy rígido en su lucha contra el fraude y si un editor hizo trampas no puede volver a participar en la plataforma [179].

Además, podría considerarse aplicar penalizaciones económicas a aquellos anunciantes que utilicen la plataforma para hacer anuncios *spam* y a todos los editores que utilicen técnicas fraudulentas con el fin de aumentar sus ingresos.

Antes de expulsar definitivamente al editor de la plataforma se concederá a las partes interesadas la oportunidad de defender su caso escribiendo una carta explicando por qué su actividad ha sido detectada como fraudulenta[8].

- **Reglas en el MIA:** Además de las políticas del MIA, se definen una serie de normas enfocadas a la prevención de fraude. Estas reglas establecen un criterio claro para expulsar del MIA a aquellos editores, anunciantes o redes que cometan fraude.

  La diferencia entre las reglas y las penalizaciones es que una infracción de las reglas conlleva la expulsión de la plataforma mientras que las penalizaciones solamente acarrean una sanción económica. Las penalizaciones son proporcionadas a cuanto se han incumplido los objetivos. Para hacer más eficiente el algoritmo se aplicarán las reglas que implican la expulsión por cada N visitas, donde $N = 1{,}000$.

  Las reglas que se han definido son:

  - **(R1) Anunciantes fraudulentos:** Para disuadir a los anunciantes de intentar mostrar anuncios de tipo *spam* se define la siguiente regla: *"Si un anunciante comete fraude en más del 20 % de sus anuncios y el número de anuncios es superior a 200 este será expulsado".*
  - **(R2) Editores fraudulentos:** El fraude por clic es uno de los temas más problemáticos en la publicidad *online*. Es necesario expulsar a los editores cuyos clics ascienden a un porcentaje p, por encima de un umbral predeterminado μ. Por lo tanto, para mantener el MIA sin fraude por clic se define la siguiente regla: *"Si un editor comete fraude en más del 20 % de sus clics y el número de clics es superior a un umbral específico, en este caso 30, entonces el editor será expulsado".*
  - **(R3) Redes fraudulentas:** Para desalentar e impedir que los editores y anunciantes de una red cometan fraude se define la siguiente regla: *"Si el 20 % o más de los anunciantes o editores de una red son fraudulentos y el número de visitas es mayor que 2.000, entonces dicha red deberá ser expulsada de la plataforma".*

[8]Este sistema es utilizado por Google y está diseñado para evitar malos entendidos. Un ejemplo de fraude mal entendido es cuando una empresa roba el código de un editor o las páginas sufren un ciberataque [180].

4.3.5.4. Módulo de selección de anuncio

Con el fin de optimizar el rendimiento del algoritmo que selecciona los anuncios se ha de definir una función que tenga en cuenta los objetivos anteriores según una métrica económica preestablecida. Debido a que el modelo tiene seis objetivos el anuncio contendrá seis variables, cada una de ellas está normalizada de tal manera que puede ser expresada como un número real en el intervalo [0,1] y donde cada objetivo está representado por su variable correspondiente.

$$\sum_{i=1}^{6} \theta_i = 1 \tag{4.3}$$

Los pesos asignados a cada variable están representados por θ_i y deben satisfacer la ecuación 4.3. Estos pesos no tienen que ser actualizados para cada visita porque esto implicaría un costo computacional muy alto. Los valores de estos pesos pueden calcularse cada varias horas o cada varios días. Además, para asegurar que los valores de los pesos son fiables, el algoritmo debe tener en cuenta un gran número de visitas. Pues un número pequeño de visitas no podría representar bien el comportamiento global de la red.

El valor óptimo de los pesos de una red puede variar dependiendo de múltiples factores como pueden ser: el número de anunciantes, el número de editores, el número de años, el promedio del fraude por clic y los anuncios de tipo *spam*. Para determinar el mejor anuncio para cada visita se asignará a cada anuncio el valor $Ranking\ Anuncio$. El valor $Ranking\ Anuncio$ se calcula para cada anuncio candidato cada vez que un usuario visita la web de un editor. Para ello, se aplica la función $F(Anuncio)$ que se expresa en 4.5. El valor $Ranking\ Anuncio$ es igual a la suma de seis números reales entre "0" y "1", por lo que estará en el rango [0,1].

$$Ranking\ Anuncio \leftarrow F(Anuncio) \tag{4.4}$$

$$\begin{aligned} F(Anuncio) = (\theta_1 \times Valor\,anuncio) + (\theta_2 \times Satisfacción\,red) \\ +(\theta_3 \times Satisfacción\,anunciante) + (\theta_4 \times Anuncios\,spam) \\ +(\theta_5 \times Costo\,campaña) + (\theta_6 \times Fraude\,editor) \end{aligned} \tag{4.5}$$

A continuación se describe cada una de las variables que representan los objetivos del MIA:

1. $Valor\,anuncio$: Representa el precio que el anunciante está dispuesto a pagar y se calcula mediante la ecuación 4.6. Cuanto más cercano sea el valor a "1", mayor será el precio que el anunciante estará dispuesto a pagar. Para normalizar el valor de esta

variable se divide el precio que el anunciante está dispuesto a pagar entre el valor máximo de la categoría.

$$Valor\,anuncio = CTR \times \frac{CPC\ Anunciante}{Max(Categoría\,CPC\ Anunciante)} \tag{4.6}$$

2. $Satisfacción\,red$: Este valor representa la satisfacción de la red mediante la relación entre los anuncios recibidos y los anuncios entregados. Este valor sirve para dar prioridad a las redes desequilibradas. Cuanto más cerca esté el valor de esta variable a "1", más satisfechos estarán los miembros de la red, tal y como se expresa en la ecuación 4.7. Por lo tanto, se trata de ayudar a las redes más insatisfechas. Los valores de las variables han sido normalizados en el rango [0,1] mediante la siguiente ecuación 4.7.

$$Satisfacción\,red = 1 - \frac{Visitas\,recibidas}{(Visitas\,recibidas + Visitas\,entregadas)} \tag{4.7}$$

3. $Satisfacción\,anunciante$: Tal y como se expresa en la ecuación 4.8, esta variable mide la satisfacción de un anunciante en función del número de impresiones de cada anunciante. Si un anunciante obtiene pocos o ninguna impresión probablemente abandone la red. Cuanto más cerca esté de "1" el valor de la variable, más insatisfecho estará el anunciante. Por lo tanto, se deben priorizar a los anunciantes que hayan mostrado pocos anuncios.

$$Satisfacción\,anun. = \frac{Visitas\,potenciales}{(Visitas\,potenciales + Visitas\,recibidas)} \times AnuncioCPC \tag{4.8}$$

4. $Anuncios\,spam$: Esta variable representa la probabilidad de que un anuncio sea de tipo *spam*. Cuantas mayores probabilidades tenga un anuncio de ser *spam*, más cercano a cero estará el valor de dicha variable. Por lo tanto, los anuncios *spam* tendrán menos probabilidades de ser mostrados.

5. $Costo\,campaña$: El precio de una campaña debe ser similar al precio general del mercado. Es decir, si un anunciante paga un precio por encima del precio de mercado, el valor de esta variable deberá acercarse a cero, tal como se expresa en la ecuación 4.9.

$$Costo\,campaña = \frac{Precio\,del\,anuncio}{(Precio\,del\,anuncio + Precio\,Real)} \tag{4.9}$$

6. $Fraude\,en\,editor$: Representa la probabilidad de que se genere un clic fraudulento. Cuanto más probable sea que el editor sea fraudulento más cercano a cero será este

valor.

4.3.5.5. Métricas en el desempeño del MIA

Con el fin de medir el rendimiento del MIA se ha establecido una métrica expresada en términos económicos. Tal y como se expresa en la ecuación 4.10 el rendimiento del MIA se puede expresar como la diferencia entre todos los ingresos del MIA y la suma de todas las penalizaciones. El algoritmo intenta maximizar los ingresos del MIA, pero al mismo tiempo intenta que se cumplan todos los objetivos con el fin de minimizar el valor de la penalización del MIA para que el rendimiento del MIA sea lo más alto posible.

$$Rendimiento\ MIA = Ingresos\ MIA - Penalizaciones\ MIA \tag{4.10}$$

Los $Ingresos\ MIA$ representan el dinero recaudado de todos los anunciantes por mostrar sus anuncios, que es igual a la suma del valor de los clics, tal y como se expresa en la ecuación 4.11.

$$Ingresos\ MIA = \sum_{j=1}^{N} Precio\ por\ clic(j) \tag{4.11}$$

Las $Penalizaciones\ MIA$ representan la suma total de todas las penalizaciones, tal y como se expresa en la ecuación 4.12.

$$Penalizaciones\ MIA = \sum_{i=1}^{5} Penalizaciones\ (i) \tag{4.12}$$

4.3.5.6. Descripción matemática del modelo

Dado un conjunto de redes publicitarias como $RPs =< RP_1, RP_2, ..., RP_n >$, con un número n de redes donde cada RP_n tiene una conjunto de anunciantes An_j de forma que $\exists An_j \in RP_n$, un conjunto de editores que $\exists Ed_k \in RP_n$ y un conjunto de visitas tal que $\exists v_l \in RP_n$.

Cada anunciante An_j se define por un conjunto de anuncios $An_j =< a_1, ..., a_m >$, donde $An_j \subseteq A$ y $(a_i \in An_j \wedge a_i \notin An_m)$, y A es el conjunto formado por todos los anuncios. Por último, V es el conjunto total de visitas $\forall v_i \in V; \forall ANs$.

El anuncio seleccionado a_i es el anuncio que pertenece al conjunto de anuncios $A =< a_1, ..., a_m >$ y también $a_i \in Ad_j$ que lleva a los ingresos máximos I.

Es decir, se selecciona $A' = \left\{a_i |\, a_i \in A' \wedge\ A' \subseteq A \therefore a_i \in A\right\}$.

Se deben maximizar los ingresos totales I_k y minimizar la suma de todas las penaliza-

ciones P_k para todos los anunciantes a_i desde RP_k, es decir, $Max\left[\sum_{k=1}^{N}(I\,{}_k^{a_i} - P\,{}_k^{a_i})\right]$ donde RP_k representa las redes, con $k =< 1, ..., N >$, por un anuncio $a_i \in An_j$ y un RP_k este modelo está sujeto a:

- $Fraude\,(a_i) > 0$: Hay fraude por parte del anunciante.
- $Fraude\,(e_i) > 0$: Hay fraude por parte del editor, donde $e_i \in E$ y E es el conjunto de editores.
- $CTR\,{}_k^{a_i} = CTR\,{}_k^{a_i} \times \Phi_j$ y Φ_j representa el número de categorías de a_i con $\Phi_j \leq p$ donde p es el número de categorías C_j y $j =< 1, ..., p > \wedge \Phi \in \mathbb{R}$.
- $CTR\,{}_k^{a_i} = F^{a_i}(x_1, x_2, \ldots, x_w)$ donde $X = (x|\, x_w)$ es una característica del anunciante a_i.
- $I\,{}_k^{a_i} = [(Clic \times CTR\,{}_k^{a_i}) \times Precio\,{}_{Clic}^{a_i}] \times tc\,{}^{a_i} - (ep\,{}^{a_i} \times M\,{}^{a_i})$ donde tc es el número total de clics en el anuncio, ep es el ingreso recibido por el editor, M es el número de muestras para los anuncios y P_z es la penalización correspondiente, y finalmente:

$$I\,{}_k^{a_i} = \sum_{i=1}^{6} \theta_i$$

4.3.6. Calcular el valor óptimo de los pesos

Aunque es fácil considerar algunos factores de la función de selección de un anuncio, se desconoce la importancia (el peso) de cada variable. Cada variable se multiplica por un peso y la suma total de todos los pesos es igual a 1, tal como se expresa en la ecuación 4.3. Para obtener el valor óptimo para todos los pesos se aplican técnicas de optimización basadas en algoritmos genéticos.

Cada vez que se produce una visita, el módulo de selección de un anuncio debe seleccionar solamente uno entre todos los candidatos. Por lo tanto, es necesario ejecutar el Algoritmo 4.1, que toma en cuenta todos los objetivos y que actualiza las variables utilizadas por la función de selección de anuncios.

En el Algoritmo 4.1 se describe el pseudocódigo para mostrar el mejor anuncio. La configuración de los pesos óptimos es la combinación que genera el mayor rendimiento del MIA según la métrica establecida. Este Algoritmo devuelve el rendimiento de MIA para una configuración de pesos.

Se podría concebir el algoritmo anterior como un pequeño módulo que devuelve el rendimiento del modelo según los pesos que se introducen como entradas. También se puede

Algoritmo 4.1 Algoritmo de modelo de intercambio de anuncios.

Entrada: ($\sum_{i=1}^{6} \theta_i = 1$: values), Datos: Anunciantes, editores y usuarios
Salida: $Fitness$

for all $v_i \in V$ **do** ▷ Para todas las visitas
  for all $a_i \in A$ **do** ▷ Para todos los anunciantes
    if $(Categoría\,(Visita) = Categoría\,(Anuncio))$ **then** ▷ Calcula valor anuncio
      $Valor(Anuncio) \leftarrow F((\theta_1 * Valor\ Anuncio) + (\theta_2 * Satisfacción\ Anunciante) + (\theta_3 * Satisfacción\ Anunciante) + (\theta_4 * Anuncios\ Spam) + (\theta_5 * Costo\,Campaña) + (\theta_6 * Fraude\ del\ editor))$
    end if
    if $(Ad\ Value > Max)$ **then** ▷ Elige al mejor anunciante
      $Máximo \leftarrow Valor_{(Anuncio)}$
      $Seleccionado\ Anuncio \leftarrow Anuncio_j$
    end if
  end for
  if $(Número\,(Visitas)\ mod\ 1000 = 0)$ **then** ▷ Si las visitas son múltiplo de 1,000.
    $p_i \in P, a_i \in A, an_i \in redes \leftarrow$ UpdateParameters() ▷ Actualiza todos los parámetros de funciones
    $AplicarR1(p_i \in P, Ad_j)$ ▷ Comprueba si hay editores tramposos.
    $AplicarR2(a_i \in A, Ad_j)$ ▷ Comprueba si hay anunciantes tramposos.
    $AplicarR3(an_i \in redes, Ad_j)$ ▷ Comprueba si hay editores tramposos.
  end if
end for
Calcula las variables: Ingresos, P_1, P_2, P_3, P_4, P_5
$Fitness \leftarrow Ingresos - (P_1 + P_2 + P_3 + P_4 + P_5)$
Devolver $Fitness$

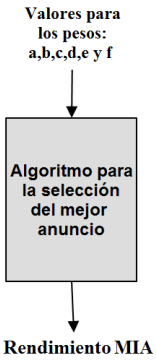

Figura 4.3: Módulo de intercambio de anuncios.

pensar en el rendimiento del modelo como el *fitness* de la función de un algoritmo genético como se muestra en la Figura 4.3 donde $\sum_{i=1}^{6} \theta_i = 1$ y $\theta_i \in R \in [0, 1]$.

4.4. Experimentos y resultados

A simple vista, se puede entender que el fraude por clic, los anuncios *spam* o la insatisfacción de ciertos anunciantes es algo que perjudica a las redes. Ahora bien, determinar de manera exacta el impacto negativo que tienen sobre una red es algo realmente complejo. Para ello, sería necesario un estudio muy detallado que analizara muchísimas variables.

Además, para realmente comprobar que este impacto es medible habría que realizar el experimento sobre una red real. Una vez que se hubiesen calculado dichos valores, no se podría asegurar que estos sean óptimos durante un tiempo largo pues en poco tiempo podría cambiar el escenario. Por otro lado, tampoco es fácil que esos valores puedan aplicarse en otra red con diferente número de visitas, de anunciantes o de editores. O incluso con otros modelos de pago.

Por lo tanto, encontrar los parámetros adecuados es algo demasiado complejo que está fuera del alcance de esta investigación. En este capítulo solamente se puede proponer un modelo que optimice los pesos a partir de un conjunto de valores, los cuales han sido asignado de forma aproximada. La ventaja de esta metodología es que los algoritmos genéticos son capaces de encontrar los valores satisfactorios para las configuraciones de los pesos.

Los valores que se han configurado de forma manual han sido los siguientes: los pesos asociados a cada penalización y que se representan con $[X_1, ..., X_5]$, los umbrales a partir de los cuales se aplican las penalizaciones y las condiciones de cada una de las regla para expulsar a un rol de la plataforma.

Respecto al valor de los pesos, en el experimento anterior todos los coeficientes se han igualado a 0,5, esto permite que el rendimiento (*fitness*) de la red no sea negativo, y que se

puedan visualizar cuales son las variables que más influyen en la decisión de un anuncio.

Los umbrales de las penalizaciones para P1, P3 y P4 representan el grado de satisfacción y han sido configurados de forma aproximada en el valor 0,25 %. Si se observa la ecuación 4.5, los valores que representan cada uno de los objetivos han sido normalizados en el rango [0,1], de forma que cuanto más alto es el valor peor es la situación y por lo tanto, mayor prioridad se le dará a este anuncio. En el caso de P2 y P5, que representan el fraude por clic y los anuncios *spam* respectivamente, se resta 0,5 veces los ingresos obtenidos por esos clics.

Respecto a los umbrales en las reglas: Se ha decidido expulsar de la plataforma a aquellas redes, editores o anunciantes que tengan más de un 20 % de anuncios fraudulentos. Para poder decidir si el usuario ha realizado fraude se necesita analizar un conjunto mínimo de muestras. Se han puesto como condiciones: En los editores, el número de clics fraudulentos debe ser mayor a 30. En los anunciantes el número de anuncios debe ser mayor a 200. Y en las redes, el número de visitas debe ser mayor a 2,000. Si en lugar de 150,000 visitas se analizaran 10 millones, estos valores podrían haber sido más altos pero el hardware disponible no permite lanzar experimentos tan costosos en tiempo.

4.4.1. Entorno de evaluación

Para realizar el experimento se han generado de forma aleatoria las visitas y la configuración de cada uno de los anunciantes, ya que no se ha encontrado ninguna base de datos con todos los campos requeridos. Para encontrar los valores óptimos de los pesos se ha aplicado un algoritmo genético.

El algoritmo genético ha sido implementado en el entorno de programación Visual Studio C# versión 12.0.31101.00 actualización 4, en un equipo con las siguientes características: Intel® Core i5-2400 CPU@3.10 GHz con 16Gb de RAM, con el sistema operativo: Windows 7 Pro, Service Pack 1 64 bits.

Se ha utilizado el paquete "Genetic Algorithm Framework" (GAF) (en español: marco de algoritmos genéticos) para C#[9] para aplicar el algoritmo genético. El paquete GAF está diseñado para implementar de forma sencilla un algoritmo genético en C#. Además, incluye una buena documentación y una gran variedad de funciones para las operaciones de cruce, mutación y selección. Además, permite la personalización de las funciones de operación por el desarrollador.

Para ejecutar nuestras pruebas se han desarrollado el entorno del MIA con la siguiente configuración: existen 10 redes de publicidad y cada una de ellas tiene 10 anunciantes y 100

[9]El GAF es un ensamblado. NET/Mono, libremente disponible a través de NuGet, que permite implementar algoritmos genéticos en el entorno de programación C# usando sólo unas pocas líneas de código [181].

editores. La probabilidad de que un editor sea fraudulento es del 18 %, para los editores es del 20 % y para las redes es aproximadamente el 10 %. La probabilidad de que un anunciante esté engañando está cerca de 15 %, pero que en el caso anterior algunos de ellos tienen una probabilidad más alta y otros una menor. Cada página de un editor puede pertenecer a 20 categorías diferentes y un anuncio puede mostrarse sólo en las páginas con la misma categoría. Para calcular los pesos óptimos se ha de ejecutar el algoritmo para las 150.000 visitas del usuario.

Los parámetros del algoritmo genético son el porcentaje de elitismo es del 5 %, el número de iteraciones en los criterios de parada es de 100, para representar los cromosomas se ha utilizado una longitud de 48. Por lo tanto, se pueden representar valores entre 0 y 2^{48}, por lo que los valores son muy altos. Con el fin de normalizar todos los valores en el intervalo [0,1] se ha dividido cada peso entre la suma total de todos los pesos.

El tamaño inicial de la población es 100, la probabilidad de mutación y la probabilidad de cruce se ejecutarán con valores de 0,1 a 1 con incrementos de 0,1. Por lo tanto, se han probado un total de 100 combinaciones diferentes tal y como se expresa en la Tabla 4.2. Para determinar la mejor combinación se elige la mejor configuración promedio después de ejecutar el algoritmo 10 veces.

El tiempo requerido para que cada ejecución se realice es de aproximadamente 14 minutos y 25 segundos. Para la operación de cruce se utiliza "Doble punto", es decir, se seleccionan dos puntos entre los que se intercambian los genes de los individuos. El método de selección de padres utilizado fue la selección mediante ruleta. Por último, el método de sustitución utilizado fue "Recambio generacional" en el que se crean nuevos individuos de padres existentes.

Una vez que se selecciona la mejor combinación, se ejecutará el algoritmo 30 veces y posteriormente se calcula el promedio, el máximo y el mínimo de la función *fitness*. También se hará un segundo experimento con la misma configuración que en el caso anterior, excepto que aplicando el valor de las penalizaciones, para ver cómo se reajustan los valores de los pesos. Los experimentos I y II han sido realizados con los siguientes números de redes: 10, 20, 30, 40 y 50.

4.4.2. Experimento I: Modelo independiente vs colaborativo

Como se ha comentado a lo largo del capítulo, cuántos más anunciantes y cuantos más editores participen mejores campañas se podrán hacer y más recursos podrán ser invertidos en mejorar la detección de fraude y en hacer mejores campañas. Además de toda la información útil que pueden intercambiar las redes para detectar el fraude.

Dicha información se refiere a cosas como técnicas fraudulentas y datos sobre las pági-

Nº de RPs	**10**	**20**	**30**	**40**	**50**
Independiente	25.149,36	50.039,76	75.402,54	100.097,97	125.197,18
Colaborativo	55.811,83	110.588,53	164.773,42	216.562,86	262.718,30

Tabla 4.1: Valores del algoritmo genético en el experimento I.

nas, anunciantes y usuarios que pueden ser útiles para combatir el fraude. Actualmente sí existen evidencias de que el *Real-Time Bidding* (RTB) tiene futuro en los modelos de intercambio de anuncio. Y de hecho se publican muchos artículos sobre RTB. No obstante, la mayoría se fija únicamente en el rendimiento económico.

Lo único que se ha hecho es comprobar el rendimiento económico. Los resultados de este experimento se reflejan en la Tabla 4.1 y también se pueden apreciar en la Figura 4.4.

En esta investigación, además de este enfoque se consideran cinco criterios más que en base a la experiencia en esta área se consideran interesantes. En este experimento se compara el rendimiento del modelo cuando las redes colaboran entre ellas y cuando las redes funcionan de forma independiente aplicando el famoso método *Generalized Second-Price* (GSP) o en español: Segundo precio generalizado. El GSP selecciona el anuncio con más valor y al anunciante se le cobra lo que ofreció el segundo anunciante con más valor. El nuevo modelo está enfocado al intercambio de anuncios entre redes por lo que no tendría sentido usarlo cuando las redes funcionan de forma independiente.

Cuando las redes son independientes, el modelo de intercambio de anuncios tiene que mostrar al usuario que visita la Red_i solamente los anuncios que pertenecen a la Red_i. En cambio, cuando las redes colaboran a este usuario se le pueden mostrar anuncios de cualquier red del modelo. Al poder elegir un anuncio entre más candidatos la solución será mejor y por lo tanto el rendimiento del modelo será mayor. Como se puede ver en los resultados, el rendimiento cuando colaboran las redes es mucho mayor.

4.4.3. Experimento II: Evaluando los pesos de las variables

En el segundo experimento se han fijado los valores de los coeficientes de cada penalización de la siguiente manera: $x_1 = x_2 = x_3 = x_4 = x_5 = 0{,}5$. Asignar a todos los pesos los mismos valores permitirá mostrar cuál es la importancia de cada uno de los objetivos cuando todos los pesos tienen el mismo valor.

La Tabla $4{,}2$ muestra los mejores valores obtenidos por el algoritmo genético para las combinaciones de cruce y mutación con incrementos de 0,1 desde 0,1 a 1. Estos valores son el promedio de 10 ejecuciones.

Los valores se calculan utilizando el valor promedio de 10 experimentos diferentes para cada probabilidad. Como se muestra en la Tabla 4.2, la mejor probabilidad de combinación

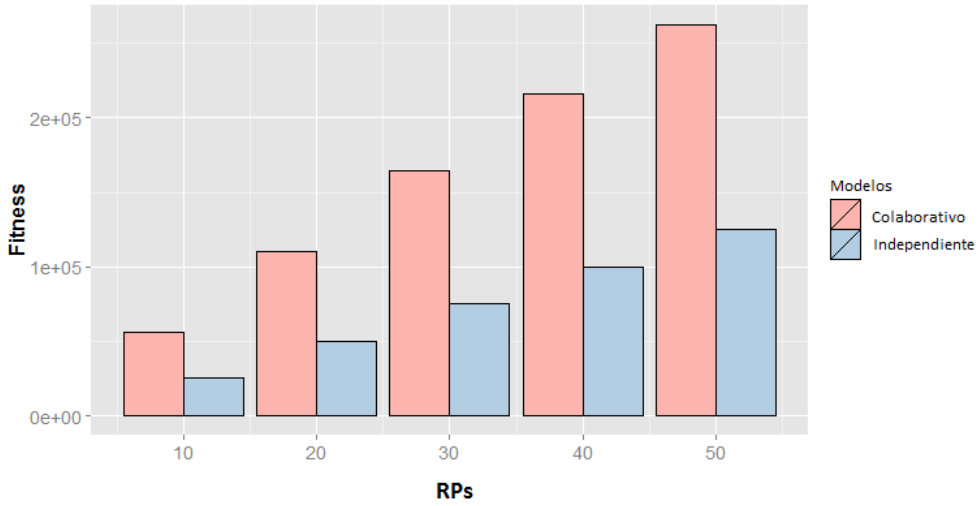

Figura 4.4: Experimento I: Modelo independiente frente a modelo colaborativo.

		Prob. crossover									
		0,1	**0,2**	**0,3**	**0,4**	**0,5**	**0,6**	**0,7**	**0,8**	**0,9**	**1**
Prob. mutación	**0,1**	9.920,4	9.971,5	9.711,7	9.997,3	9.783,6	9.763,8	10.018,1	9.893,0	9.750,1	9.898,5
	0,2	9.016,2	9.538,9	9.761,9	9.753,2	9.737,0	10.012,7	**10.032,3**	9.532,8	9.775,0	9.785,7
	0,3	9.509,8	9.757,2	9.810,6	9.630,8	9.804,0	9.808,1	9.493,8	9.803,0	9.693,1	9.606,2
	0,4	9.761,3	9.819,3	9.756,6	9.920,3	9.687,9	9.547,6	9.844,0	9.443,6	9.549,6	9.755,2
	0,5	9.828,0	9.561,0	9.625,4	9.454,0	9.633,1	9.710,0	9.743,5	9.873,1	9.365,4	9.629,7
	0,6	9.717,2	9.813,5	9.310,7	9.730,9	9.430,4	9.929,8	9.761,7	9.525,6	9.436,9	9.671,4
	0,7	9.507,1	9.604,4	9.569,9	9.691,2	9.565,6	9.490,1	9.532,3	9.878,3	9.297,7	9.255,0
	0,8	9.932,8	9.776,1	9.212,0	9.417,7	9.513,3	9.724,2	9.738,0	9.312,8	9.410,1	9.825,9
	0,9	9.681,5	9.383,4	9.490,5	9.732,4	9.708,5	9.691,3	9.755,8	9.454,7	9.534,1	9.532,3
	1	9.609,7	9.479,9	9.788,1	9.716,4	9.630,7	9.609,4	9.977,5	9.383,0	9.893,3	9.947,2

Tabla 4.2: *Fitness* para cada valor de cruce y mutación.

se compone de una probabilidad de *cross-over* de 0,2 y una probabilidad de mutación de 0,7.

Una vez que se calcula la mejor combinación, se ejecuta el algoritmo 30 veces, se calcula el promedio y se obtendrán los resultados mostrados en la Tabla 4.3. También se puede ver de manera gráfica en la Figura 4.5.

Para ver que este modelo es bueno y que tiene sentido, se ha comparado el rendimiento de este modelo con el modelo GSP. Después de aplicar el GSP, se aplicarán las penalizaciones que se han considerado que tiene este modelo. Como se puede apreciar, el método GSP no tiene en cuenta ningún objetivo de la red salvo el rendimiento económico y es por ello por lo que obtienen tantas penalizaciones que hacen que el rendimiento del modelo sea tan bajo.

Esto hace pensar que este modelo si es interesante para aquellas redes que quieran tener

Nº de RPs	10	20	30	40	50
AG	10.146,59	19.188,95	30.861,78	41.587,55	50.167,97
GSP	-26.727,29	-41.331,81	-63.645,60	-100.379,85	-124.853,94

Tabla 4.3: Valores del algoritmo genético vs modelo GSP.

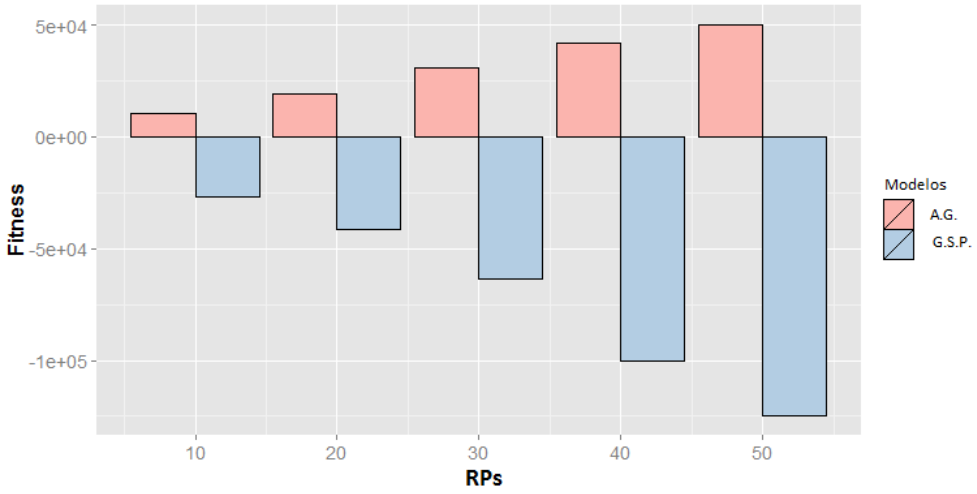

Figura 4.5: Experimento II: Rendimiento de los modelos AG y GSP.

un buen funcionamiento y que vean con buenos ojos que todos los roles estén satisfechos.

Los valores óptimos de los coeficientes para la función *fitness* se muestran en la Figura 4.6. Se han ordenado las variables en orden descendente según su importancia. Cada peso tiene un valor en el rango [0,1] que refleja la importancia de cada uno de los objetivos. También se han representado con una línea discontinua de color rojo el promedio de todas las variables.

Como se puede apreciar claramente, los pesos θ_5 y θ_2 son los más importantes. Se tiene que tener en cuenta que la métrica utilizada en la función *fitness* se define en términos económicos. El peso θ_5 se asocia con la variable $Costo\,Campaña$ e indica si la campaña de un anunciante ha tenido un costo por encima del precio de mercado. Si los anunciantes que están dispuestos a pagar más dinero por un anuncio abandonasen la plataforma es probable que los ingresos caigan dramáticamente.

Por otro lado, θ_2 representa el peso que regula la variable $Satisfacción\ Red$ que representa la satisfacción de la red respecto al balance del número de visitas recibidas y entregadas. Si una red deja de participar en el MIA, se perderán todos los editores y todos los anunciantes que pertenecen a dicha red, por lo que los costos serían muy grandes.

θ_1 representa el peso asociado a la variable $Valor\,del\,anuncio$ que indica el valor del anuncio. Es lógico que tenga un valor alto porque si se seleccionan los anuncios más renta-

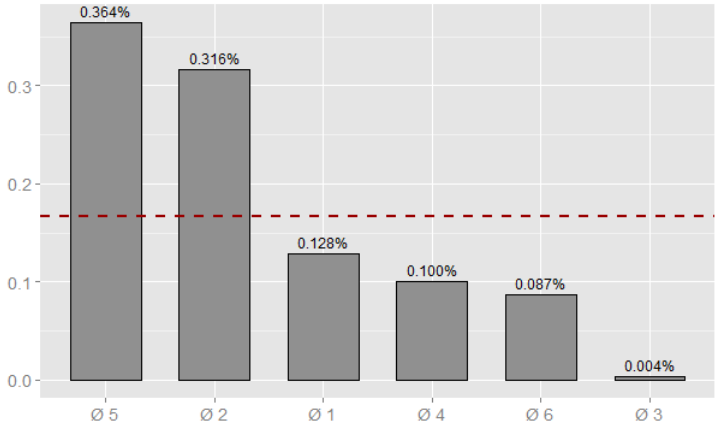

Figura 4.6: Experimento II: Mejor configuración de los pesos.

bles, los ingresos de las redes aumentarán.

Los pesos θ_4 y θ_6 reflejan los valores asociados a fraude. El peso θ_4 se asocia con la variable $Spam\ Adverts$ que indica la probabilidad de que un anuncio sea de tipo *spam.* El peso θ_6 se asocia con la variable $Fraude\ Editor$ e indica la probabilidad de que un anuncio sea fraudulento.

Mostrar anuncios de tipo *spam* y recibir clics fraudulentos tiene un impacto negativo en el MIA y por lo tanto, el valor de estos dos pesos debe ser similar.

Finalmente θ_3 se asocia con la $Satisfacción\ del\ anunciante$ que indica la satisfacción con respecto al número de anuncios mostrados. Este peso generalmente tiene un valor cercano a cero y lleva a pensar que casi no tiene importancia, ya que el peso θ_5 cumple esta función de forma indirecta. Esto significa que si las redes se equilibran, es probable que el número de anuncios publicados por los anunciantes se vean equilibrados.

4.4.4. Experimento III: Evaluando la adaptabilidad del modelo

En los resultados del experimento II, θ_3 es el que tiene menos peso en la optimización de cada objetivo. En el siguiente experimento se aumentarán los pesos asociados con el objetivo 3 para comprobar que el algoritmo genético es capaz de adaptarse a estos cambios.

Para lograr este propósito, se han creado un experimento en el que solamente se cambia el peso de las penalizaciones con los siguientes valores: $x_1 = x_2 = x_4 = x_5 = 0{,}5$, mientras que $x_3 = 3$, representa el valor asociado a la variable θ_5**.**

El objetivo de este experimento es comprobar que el algoritmo genético es capaz de ajustar el valor de sus pesos para adaptarse a la nueva configuración de modelo. El resto de los parámetros conserva la configuración del experimento II.

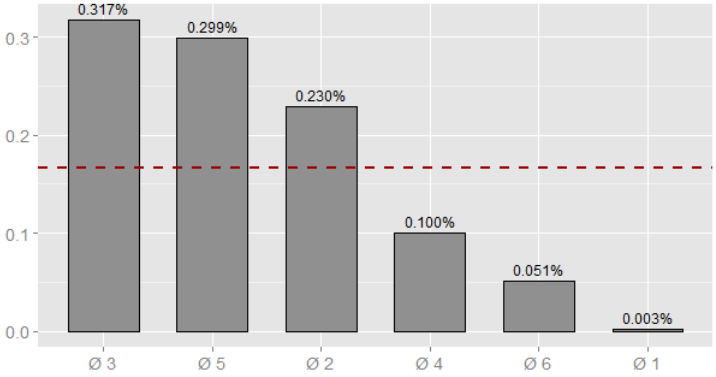

Figura 4.7: Experimento III: Mejor configuración de los pesos.

En este modelo se utiliza la misma configuración que en el modelo anterior, y también se muestran los valores promedio que se calcularon.

La Figura 4.7 muestra los resultados de la mejor configuración de pesos en la nueva configuración. Como se muestra en la Figura 4.7, el valor más importante es θ_3 y representa la satisfacción del anunciante.

Se puede observar que los valores de θ_5, θ_2, θ_4 y θ_6 siguen manteniendo en cuanto a sus pesos el mismo orden que tenían en la Figura 4.7. Esto se debe a que se ha cambiado el valor de una sola variable. Sin embargo, θ_1 se redujo mucho. Esto se debe al hecho de que la suma de los coeficientes de las penalizaciones en el experimento II es $x_1+x_2+x_3+x_4+x_5 = 2{,}5$, y en el experimento III es de $x_1 + x_2 + x_3 + x_4 + x_5 = 5$. La conclusión es simple, se ha comprobado que si se cambian los valores de las penalizaciones, los valores de los pesos también cambian. De forma que la función de selección de anuncios queda nuevamente optimizada.

4.5. Conclusiones

La mayoría de los anunciantes en Internet han optado por grandes redes publicitarias porque son capaces de ofrecer campañas más rentables. Además, la mayoría de los editores se asocian con grandes plataformas porque obtienen mayores ingresos y porque reciben sus pagos puntualmente.

Tener más anunciantes y más editores hace que las grandes redes obtengan mayores ingresos, lo que les permite ofrecer a los anunciantes y a los editores más servicios y de mayor calidad. Además, estas redes fácilmente pueden invertir en la promoción de su plataforma y en el reclutamiento de nuevos anunciantes y editores.

En este capítulo se ha desarrollado una metodología que no ha sido publicada con anterioridad. Se debe tener en cuenta que la mayoría de las redes de publicidad no revelan sus algoritmos, ya que esto significaría perder parte de su ventaja competitiva que implica muchos años de investigación.

La metodología desarrollada sirve para seleccionar un anuncio en un MIA. A lo largo del capítulo, se ha podido ver que la selección de un anuncio es una tarea compleja que debe tomar en cuenta múltiples objetivos, a menudo con intereses enfrentados, y en donde cada objetivo se asocia con un peso para ser optimizado.

Uno de los principales logros de esta investigación es haber dado un punto de partida desde la cual puede construirse un MIA que tome en cuenta las principales amenazas y problemas de la publicidad *online*. Además, se ha desarrollado una metodología para encontrar los mejores pesos definiendo un *fitness* que considera todos los objetivos necesarios para crear un buen ecosistema para el MIA.

El objetivo de la tesis no es tanto desarrollar una nueva técnica que mejore la predicción del CTR o la detección de fraude, sino desarrollar una metodología que ayude a seleccionar el mejor anuncio asumiendo que el CTR y los módulos de detección de fraude han sido desarrollados correctamente. Obviamente, cuanto más confiables y exactos sean los módulos que estimen el CTR o detecten el fraude, mayor será el rendimiento del modelo.

Se ha comprobado que los pesos óptimos para el módulo de selección de un anuncio varían dependiendo de la importancia que tengan los objetivos, del valor de las penalizaciones, del número de anunciantes y campañas, así como de la configuración de todo lo que compone el MIA. Por lo tanto, se puede afirmar que no existe una configuración óptima que pueda extenderse a todos los modelos. Lo que sí se podría aplicar a todas las redes es la metodología desarrollada para configurar la función para la selección de un anuncio, de forma que el rendimiento de la redes pueda ser maximizado en base a los objetivos definidos. El rendimiento respecto al GSP ha sido superior, principalmente porque el modelo GSP no tiene en cuenta las penalizaciones asociadas a cada objetivo sino únicamente el rendimiento económico.

Trabajar con datos reales es algo muy complicado ya que, aunque se puede contar con algunas bases de datos de información, es prácticamente imposible encontrar una base de datos que contenga todos los campos requeridos por este algoritmo.

Otro factor que hemos de tener en cuenta, es que el modelo propuesto está enfocado al medio/largo plazo. Los anunciantes, editores y redes asociadas no van a notar la mejora del nuevo modelo de un día para otro, sino que se necesitan algunas semanas para que los editores vean que no se publican anuncios *spam* en sus páginas, que los anunciantes se den cuenta de que sus campañas dan buen resultado y de que las redes que colaboran tomen la

decisión de quedarse. La satisfacción de cada uno de los roles hará que estos permanezcan y que los nuevos roles que participen también quieran quedarse.

Finalmente, para que la metodología propuesta sea viable es necesario que el tiempo de respuesta del algoritmo sea de escasas décimas de segundo. Para ello, habría que analizar el hardware y el software que implementara el algoritmo. Podría ser apropiado aplicar la paralelización y también una arquitectura distribuida. De forma que el modelo se pudiese replicar y así balancear la carga de consultas. Además, los distintos modelos deberían de sincronizarse y para ello, se debería crear un protocolo de comunicación.

Capítulo 5

Diseño de los módulos implicados en el cálculo del valor de un anuncio mediante métodos supervisados de *machine learning*

5.1. Introducción

En el capítulo anterior solamente se utiliza el modelo de pago CPC. Debido a que los modelos de pago CPM y CPA también son muy utilizados en la publicidad en internet, se propone una metodología para el cálculo del valor de un anuncio en las redes CPM, CPC y CPA. Esto se llevará a cabo añadiendo un nuevo módulo que calcule la probabilidad de que se genere una venta y considerando la salida del módulo que establece probabilidad de fraude. En la Figura 5.1 se muestran los módulos que se utilizan para calcular el valor de un anuncio.

Estimar con precisión este valor aumenta los ingresos de estas tres redes pues permite seleccionar el anuncio más rentable. Al aumentar los ingresos, se puede pagar mejor a los editores y se pueden desarrollar mejores servicios para los anunciantes. Para el desarrollo de esta metodología se propone un sistema basado en métodos tradicionales de *aprendizaje automático* y en métodos *Deep Learning*.

En este capítulo se propone una metodología para determinar el valor de un anuncio en las redes CPM, CPC y CPA [182], que son las formas de pago más extendidas hoy en día.

El valor de un anuncio se puede expresar como los ingresos en dólares que genera para la red publicitaria mostrar un anuncio en la página web de un editor. Cada modelo de pago cobra en función de diversos factores y cada modelo tiene su propia fórmula para calcular

el valor del anuncio. También se ha de tener en cuenta que los anuncios pueden ser *spam* [183]. En los anuncios de tipo *spam* el anunciante intenta engañar a los usuarios para obtener ingresos, información privada o para instalar un programa malicioso en su ordenador sin su aprobación. Por lo tanto, son muy perjudiciales para el ecosistema publicitario y pueden provocar importantes pérdidas económicas.

Calcular de manera efectiva el valor de un anuncio aumenta los ingresos de la red publicitaria debido a que puede mostrar el anuncio más rentable de todas las campañas [32]. Si la red publicitaria aumenta sus ingresos, los editores recibirán más ingresos y, por ende, más editores estarán dispuestos a unirse a la plataforma [184].

Esta metodología también propone a los anunciantes el método de pago más beneficioso para sus campañas entre el CPM, el CPC y el CPA [185]. Cada anunciante también puede elegir el método de pago de su preferencia [186]. Esto aumenta la satisfacción de los anunciantes y permite a los anunciantes utilizar diferentes modelos en la misma plataforma.

Para aplicar la metodología para el cálculo del valor de un anuncio se ha desarrollado un sistema compuesto por cuatro módulos. El Módulo 1 calcula la probabilidad de que el anuncio sea *spam*. El Módulo 2 calcula el CTR, que es la probabilidad de que un usuario acceda a la página web de un editor y de que se genere un clic en un anuncio [54]. El Módulo 3 calcula la probabilidad de que se genere una venta a partir de una visita. Por último, el Módulo 4 calcula el valor de un anuncio según el modelo de pago escogido por el anunciante y el precio que está dispuesto a pagar.

Los beneficios de las empresas de publicidad *online* han aumentado considerablemente en los últimos gracias a los métodos de ML [187]. A diferencia de los métodos estadísticos clásicos, los métodos ML pueden manejar la incertidumbre. Algunos algoritmos de clasificación usan una función de distribución de Bernoulli para la predicción de eventos que depende de otros factores. Los métodos de ML pueden modelar comportamientos sin una relación lineal entre las variables independientes.

Gracias a estas técnicas, los eventos pueden ser modelados a pesar de la incertidumbre y de la independencia de estos eventos con otras variables del modelo. Además, las técnicas para predecir eventos permiten tratar con características como la ciclicidad, la estacionalidad y las tendencias.

Debido a todas estas razones se han utilizado los métodos de ML supervisados para aplicar los módulos 1, 2 y 3. Para los módulos 1 y 2, se han utilizado métodos de clasificación y para el módulo 3 se han utilizado métodos de regresión. Con el fin de encontrar el mejor modelo se han utilizado métodos tradicionales de ML que ya están implementados en el paquete Caret de R Studio[1] [190]. Teniendo en cuenta la mejora de los métodos *Deep Lear-*

[1]El paquete Caret incorpora un conjunto de funciones para crear y evaluar modelos predictivos [188]. R

ning (DL) respecto a los métodos tradicionales de ML, también resulta interesante aplicar los métodos DL en esta investigación.

DL tiene una arquitectura profunda, lo cual hace referencia al número de niveles o capas necesarios para construir una función no lineal [191]. Además, los métodos DL funcionan mejor con datos muy complejos [192] que son precisamente el tipo de datos que se está tratando.

En la segunda sección del capítulo se presentan varios estudios relacionados con el cálculo del valor de un anuncio. También se presentan los métodos de selección de variables utilizados en nuestros experimentos: RFE [193] y DL. En la tercera sección se presenta una metodología para calcular el valor de un anuncio en las redes CPM, CPC y CPA. En la cuarta sección se describen la base de datos y los resultados obtenidos. Y en la quinta sección se presentan los resultados y una propuesta para trabajos futuros.

5.2. Cálculo del valor de un anuncio

Las redes CPM [194] se utilizaron en los inicios de la publicidad *online* y consisten en pagar una cantidad C cada vez que el anuncio a_i se visualiza 1.000 veces. En este modelo de pago la rentabilidad es muy sencilla de calcular, ya que el precio depende sólo del número de veces que el anuncio es mostrado. El valor de un anuncio es igual al costo CPM dividido entre mil, tal y como se expresa en la ecuación 5.1.

$$CPM\,(a_i) = \frac{CPM}{1000} \tag{5.1}$$

La redes CPC son las más utilizadas en la actualidad debido a que lo han elegido las mayores empresas de publicidad *online* [195]. En las redes CPC el anunciante paga únicamente cuando un usuario hace clic en un anuncio. Por lo tanto, se necesita un modelo de ML supervisado para calcular la probabilidad de que un usuario genere un clic en función de algunos parámetros sobre el usuario y de la página web en la que el anuncio a_i se visualiza. El valor del anuncio se calcula como el precio por clic que el anunciante está dispuesto a pagar multiplicado por el CTR, tal y como se expresa en la ecuación 5.2.

$$CPC\,(a_i) = CTR \times Precio\,Clic \tag{5.2}$$

La forma de pago en las redes CPA consiste en pagar a los editores cuando se produce una conversión[2]. Se necesita un modelo supervisado de ML para calcular la probabilidad de

Studio es un entorno de desarrollo integrado orientado a computación y gráficos estadísticos[189].

[2] Una conversión es una acción realizada por el usuario y se produce cuando un usuario que visita la página

generar una conversión. El valor de un anuncio a_i en este modelo se calcula como el CTR multiplicado por la probabilidad de que se genere una conversión y multiplicado por el valor de una comisión, tal como se expresa en la ecuación 5.3.

$$CPA\,(a_i) = CTR{\times}Comisión \times P(Comisión) \tag{5.3}$$

Todas las fórmulas anteriores se utilizaron para calcular el valor de un anuncio mediante los tres métodos de pago anteriores. Se tiene que considerar que un anuncio puede ser de tipo *spam*. Por lo tanto, se debe multiplicar el anuncio por la probabilidad de que no sea *spam*, es decir, $1 - P(spam)$. De tal forma que cuanto mayor sea la probabilidad de que un anuncio sea *spam* menor será el valor del anuncio.

Se ha diseñado un sistema para aplicar la metodología descrita en la Figura 5.2. Este sistema tiene como entradas la visita, el método de pago y la cantidad de pago que el anunciante está dispuesto a pagar. La salida del modelo consiste en el valor del anuncio expresado en dólares.

Este sistema de la Figura 5.1 está compuesto por cuatro módulos independientes. El módulo 1 calcula la probabilidad de que un determinado anuncio a_i sea *spam*, $P(a_i|spam)$. El módulo 2 calcula la probabilidad de que el anuncio a_i reciba un clic, es decir: $P(a_i|clic)$. El módulo 3 estima la probabilidad de que un producto sea vendido a partir de un clic, es decir, $Fc(Ventas \mid P(a_i|clic))$. Las entradas de los módulos 1, 2 y 3 son atributos sobre el usuario y la visita. El módulo 4 calcula el valor de un anuncio basado en el modelo de pago y el precio establecido por el propietario.

5.2.1. Módulo 1: Cálculo de la probabilidad de *spam*

Este modelo de clasificación supervisada tiene dos posibles clases: (*spam, no spam*), y predice la probabilidad de que el anuncio sea *spam* en el rango [0,1]. Esta probabilidad se multiplica por el valor de un anuncio en las redes CPM, CPC y CPA de manera que cuanto mayor sea la probabilidad de *spam* menor será el valor de un anuncio.

Cuando se muestra un anuncio de tipo *spam* tiene un efecto negativo sobre la red[3]. Este

web de un editor, hace clic en el *banner* del anunciante y por último, realiza una acción acordada previamente con el anunciante, como una compra o rellenar un formulario.

[3] Este tipo de anuncios pueden redirigir a los usuarios a una página con un virus. Estos mensajes se conocen como *badvertiser* [196]. También puede redirigir a los usuarios a una página web con contenido ilegal [40] o a una página web con un *script* que intenta estafar a los usuarios a través de *phishing* [197].

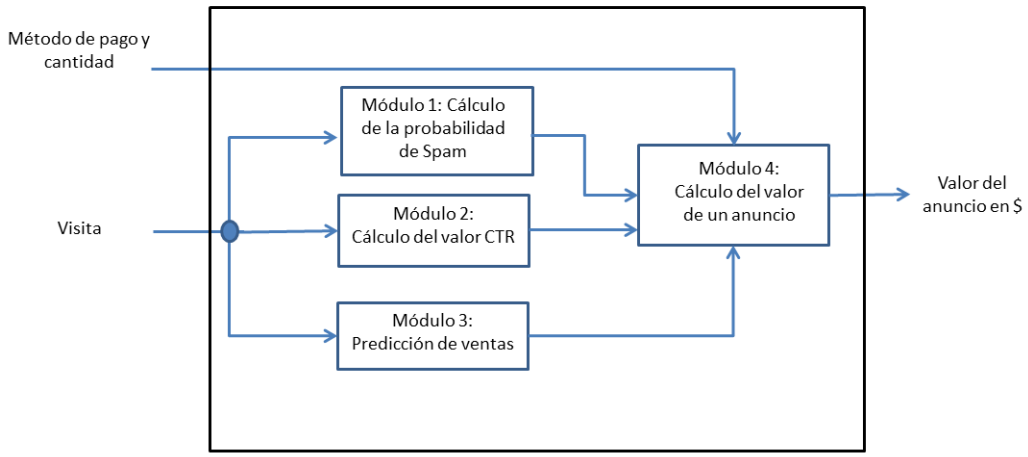

Figura 5.1: Sistema para calcular eficazmente el valor de un anuncio.

tipo de anuncios daña la reputación de la plataforma de publicidad ya que disuade a muchos usuarios de hacer clic en los *banners*. Además, puede causar un perjuicio a las marcas que aparecen en estas plataformas.

La probabilidad P de que un anuncio x_i sea *spam*, $P(x_i|spam)$, puede expresarse como $P(x_i|spam) = \alpha$ y $P(x_i|no\ spam) = 1-P(x_i|spam)$. Los eventos para evaluar si un anuncio es *spam* siguen una distribución Bernoulli con clase C, donde $C = \{spam,\ no\,spam\}$, y X es un conjunto discreto. En este módulo se construye un modelo discriminante conocido como clasificador binario r_i^t, mediante una regresión logística, donde $X = \{x^t, r^t\}_{t=1}^N$ y

$$r_i^t = \begin{cases} 1 & x^t \in C_i \\ 0 & x^t \in C_j \end{cases} \quad for\ j \neq i$$

Por lo tanto, la estimación de la probabilidad de que un anuncio sea *spam* se puede obtener mediante la ecuación 5.4.

$$P(x^t|spam) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}} \tag{5.4}$$

Donde la probabilidad de *spam* se puede calcular mediante la ecuación 5,5.

$$P(x^t|no\ spam) = 1 - P(x^t|spam) \tag{5.5}$$

Linealizando con la transformación $Y = \frac{lny}{(1-y)}$ se obtiene la ecuación 5.6.

$$\frac{\log P(x^t)}{1-P(x^t)} = \alpha + \beta x \tag{5.6}$$

y así queda demostrado que el modelo es lineal.

5.2.2. Módulo 2: Cálculo del valor CTR

De igual forma que en el módulo anterior, el modelo supervisado de clasificación del CTR tiene dos posibles clases: $(clic,\ no\,clic)$, y estima el CTR de un anuncio, de tal manera que cuanto mayor sea la probabilidad de que un usuario genere un clic mayor será la probabilidad de que se muestre. El CTR es utilizado en la ecuación del valor de un anuncio en las redes CPC y CPA.

Para el módulo CTR se crea un clasificador binario $r\,_t^i$ utilizando regresión logística en el que $X = \{x^t, r^t\}_{t=1}^N$ y con sólo dos posibles clases C, donde $C = \{clic,\ no\,clic\}$. El conjunto X es un conjunto discreto y

$$r_i^t = \begin{cases} 1 & x^t \in C_i \\ 0 & x^t \in C_j \end{cases} \quad for\ j \neq i$$

La probabilidad estimada de que se genere un clic se calcula con la ecuación 5.7.

$$P(x^t|clic) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}} \tag{5.7}$$

Por lo tanto, la probabilidad de que no se genere ningún clic se representa con la ecuación 5.8.

$$P(x^t|no\,clic) = 1 - P(x^t|clic) \tag{5.8}$$

Linealizando mediante la transformación $Y = \ln \frac{y}{1-y}$ se tiene la ecuación 5.9.

$$\frac{\log P(x^t)}{1-P(x^t)} = \alpha + \beta x \tag{5.9}$$

Por lo tanto, queda demostrado que el modelo está linealizado.

5.2.3. Módulo 3: Predicción de ventas

Los modelos supervisados de regresión calculan el número de ventas V a partir de una base de datos proporcionada por el cliente. La variable V se usa en la ecuación del método de pago CPA. Este modelo estima una regresión lineal multivariable donde X son las variables explicativas y V es el número de ventas estimado.

V será determinado mediante $r\,_t^i$ y $r\,_t^i = g(x^t, z^t)$, donde z^t equivale a la desviación causada por las variables escondidas sin considerar el error de rendimiento. Esto lo se expresa mediante la ecuación de regresión lineal 5.10.

$$V = r^t = \beta_0 + \beta_0 x_1 + \beta_2 x_2 + ... + \beta_n x_n \tag{5.10}$$

Para calcular el precio de una visita se consideran las siguientes variables:

- Y: Número de ventas.
- C: Comisión por venta.
- X: Número de clics.

Donde el precio de un clic se calcula mediante:

- $Precio\,de\,un\,clic = (Y \times C)/X$
- $Precio\,de\,una\,visita = CTR \times Precio\,Clic \times (1 - P(Spam))$

Se necesita cierta información de los clientes. Por ejemplo, se necesita saber el dinero que gastan en productos *online* antes y después del lanzamiento de la campaña. También se requiere otra información como la descripción del producto, la categoría del producto y el descuento asociado al producto.

No todos los anunciantes comparten esta información porque si se llegara a revelar información confidencial podrían poner a sus empresas en riesgo de pérdidas económicas. La mayoría de los anunciantes dan una comisión al editor sólo cuando el usuario genera un clic en el *banner* del editor en un período de 15-30 días antes de la compra.

En estos casos, los *banners* tienen un *script* que deja una *cookie* en la memoria caché del usuario [198]. También puede haber una comisión ganada simplemente porque el usuario ha visto un *banner*, que deja un tipo de *cookie* diferente. Este último, no es muy común y el valor de la comisión es menor.

5.2.4. Módulo 4: Cálculo del valor de un anuncio

En el módulo 4 no es necesario un modelo supervisado de ML. Para calcular el valor de un anuncio en cada modelo se aplica una ecuación diferente. Este módulo tiene cuatro entradas. Tres de ellos son las salidas de los módulos 1, 2 y 3, y la cuarta entrada indica el tipo de modelo y el precio establecido por el anunciante.

Para calcular el valor de un anuncio en las redes CPM sólo se necesita la salida del módulo 1 como se muestra en la ecuación 5.11.

$$CPM\ (Valor\ anuncio) = (CPM/1000) \times (1 - P(Spam)) \tag{5.11}$$

Para calcular el valor de un anuncio en las redes CPC es necesario la salida de los módulos 1 y 2, como se muestra en la ecuación 5.12.

$$CPC\,(Valor\ anuncio) = CTR \times Precio\,Clic \times (1 - P(Spam)) \tag{5.12}$$

Y para calcular el valor de un anuncio en las redes CPA se necesita la salida de los módulos 1, 2 y 3. A partir del número de ventas se puede calcular el valor de un clic usando la ecuación 5.13. El precio medio de un clic puede calcularse como el número de ventas multiplicado por el valor de la comisión y dividido entre el número total de clics. El valor de un anuncio se puede calcular con las siguientes ecuaciones 5.13 y 5.13.

$$Valor\ Clic = \frac{Número\ de\ ventas \times Comisión\ Precio}{Número\ de\ clics} \tag{5.13}$$

$$CPA\,(Valor\ anuncio) = CTR \times Valor\ Clic \times (1 - P(Spam)) \tag{5.14}$$

La ecuación 5.15 se puede aplicar para seleccionar qué anuncio mostrar.

$$Ranking\ Anuncio = Valor\ Anuncio \times Valor\ Calidad \tag{5.15}$$

$Ranking\,Anuncio$ es la multiplicación entre el valor de un anuncio $Valor\,Anuncio$ y su calidad $ValorCalidad$. $Ranking\,Anuncio$ se recalcula cada vez que un anuncio se muestra. El $Valor\ Anuncio$ corresponde a la salida del sistema y representa el valor del anuncio expresado en dólares. No siempre se muestra el anuncio más rentable, de lo contrario, un anuncio cuyo valor estimado sea menor que el de otra campaña con la misma configuración nunca será mostrarlo. Por lo tanto, se deben tener en cuenta muchos otros factores y por esta razón se utiliza el $Valor\ Calidad$.

$Valor\,Calidad$ es una variable que se calcula teniendo en cuenta los factores necesarios para que pueda desarrollarse adecuadamente un ecosistema publicitario. Algunos de estos factores incluyen el rendimiento de la campaña, la relación entre los anuncios mostrados y todas las posibles muestras de anuncios, la satisfacción de los usuarios que visitan la página web del anunciante, el número de días de la campaña y otros factores.

El cálculo del valor de un anuncio, también puede tener un efecto sobre los beneficios del cliente. Se consideran las siguientes variables en el cálculo del punto de equilibrio para cada uno de los modelos:

- C: El costo fijo de desarrollo de productos.
- Z: El precio de venta del producto.
- Cl: El número de clics.
- N: El número de transacciones.

- V: El número de visitantes por página.
- M: El número de muestras.
- Y: El costo de la comisión.
- W: El costo del anuncio.
- $Beneficio$: El beneficio obtenido por la red.
- $Punto\ de\ equilibrio(PE)$: El PE entre las ganancias y las pérdidas.

Por lo tanto, se definen las siguientes ecuaciones:

$$Beneficio = Z - C(Por\ transacción) \tag{5.16}$$

$$Beneficio\ Neto = Beneficio \times V \tag{5.17}$$

$$CTR = (Cl \times V)/M \tag{5.18}$$

Donde el Punto de equilibrio (PE) se calcula con las ecuaciones:

$$PE(CPM) = (Beneficio\ Neto/V)/1000 \tag{5.19}$$

$$PE(CPC) = Beneficio\ Neto/V \tag{5.20}$$

$$PE(CPA) = (Beneficio\ Neto–Y)/V \tag{5.21}$$

$$Beneficio\ CPM\ sin\ costo = Beneficio – (PE(Valor\ CPM) \times V) \tag{5.22}$$

$$Beneficio\ CPC\ sin\ costo = Beneficio – (PE(Valor\ CPC) \times V) \tag{5.23}$$

$$Beneficio\ CPA\ sin\ costo = Beneficio – (PE(Valor\ CPA) \times V) \tag{5.24}$$

5.3. Experimentos y resultados

5.3.1. Entorno para los experimentos

Para los experimentos se ha utilizado la versión 3.1.2 (2014-10-31) del software R Studio. Este software se ha ejecutado sobre un equipo Intel®Core i5-2400 CPU@3.10 GHz con 16 Gb de RAM y con el sistema operativo Windows 7 Pro Service Pack 1 64 bits.

Para aplicar la metodología descrita en la Figura 5.2 se han utilizado métodos supervisados de ML en los módulos 1, 2 y 3, con la excepción de los modelos DL mediante el paquete

Caret [199]. Para la construcción de los modelos DL se ha utilizado el paquete H2O DL de R Studio [200] y se han usado las *Deep Learning neural networks* (DLNNs).

Por último, se considera que el *Root Mean Squared Error* (RMSE)[4], es una métrica adecuada para evaluar la precisión de los modelos generados. El RMSE mide el promedio de los errores al cuadrado, de tal manera que el modelo con menor RMSE será el modelo seleccionado. La fórmula del RMSE puede expresarse con la ecuación 5.25.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(X_{pred\,i} - X_{model\,i})^2}{n}} \tag{5.25}$$

Donde X_{pred} es el conjunto de los valores predichos y X_{model} es el conjunto de valores del equipo de prueba. La base de datos para los módulos 1 y 3 contiene valores vacíos. Estos valores se sustituyen por otros aplicando la función *"ReplaceMissingValues"* del software WEKA[5]. Esta función reemplaza los valores que faltan con atributos numéricos que se calculan como el promedio o la moda del resto de los valores de los atributos.

Para seleccionar el conjunto óptimo de las variables, en primer lugar, se eliminan las variables irrelevantes a través de un filtro de R Studio llamado *"NearZeroVar"*. Este filtro detecta los atributos con poca o ninguna variación y analiza las variables de forma independiente, es decir, sin la construcción de cualquier modelo. Por lo tanto su ejecución es mucho más rápida que en los métodos *wrapper*.

5.3.2. Implementación del algoritmo *Deep Learning*

Algunos autores pueden argumentar que los métodos DL necesitan mucho tiempo para ser construidos. Aunque esto puede ser cierto, se deben considerar otros dos factores. En primer lugar, la potencia de cálculo aumenta cada año gracias a la paralelización. Y en segundo lugar, el cuello de botella no se encuentra tanto en el proceso de construcción del modelo, que puede no hacerse en tiempo real, sino en el tiempo de respuesta a una entrada. Y una vez que se construye el modelo, el tiempo de respuesta debe ser inferior a un umbral.

Para esta investigación se ha utilizado el paquete H2O[6] de R Studio. El paquete H2O implementa una red neuronal artificial multicapa con *feedforward* formada con descenso de gradiente estocástico usando *"Propagación hacia atrás"* en las capas ocultas.

Las aplicaciones de estos algoritmos pueden variar desde la detección de fraude hasta la predicción del rendimiento en el mercado de valores [180]. También se utilizó una red

[4]RMSE es ampliamente utilizado para los modelos de regresión y que se define por la raíz cuadrada de la media de la diferencia entre la casa de las muestras y el valor esperado de estas muestras.

[5]Weka es un software de minería de datos en Java, implementa una colección de algoritmos de aprendizaje automático para de minería de datos[201].

[6]El H2O DL paquete es una plataforma de código abierto para la memoria de aprendizaje automatizado

Algoritmo 5.1 Algoritmo implementado en Caret para la selección del mejor modelo.

```
1: Aplicar RFE al la base de datos
2: Definir los métodos que se quieren evaluar
3: for (Cada método) do
4:     Dividir la base de datos en el conjunto de pruebas y el conjunto de entrenamiento
5:     Definir el conjunto de configuraciones para evaluar el valor de los parámetros
6:     for (Cada conjunto de parámetros) do
7:         for (Para cada muestra generada del conjunto de datos de entrenamiento) do
8:             Elegir y separar unas muestras para el test
9:             Crear el modelo con las muestras restantes
10:            Predecir las muestras seleccionadas para el test
11:        end for
12:        Calcular el promedio del rendimiento para todas las predicciones realizadas
13:    end for
14:    Determinar la configuración óptima de los parámetros
15:    Predecir el valor para la base de datos de pruebas
16: end for
17: Determinar el mejor método y su configuración
```

neuronal artificial con *feedforward* multicapa para todos los modelos supervisados. Se configuró una red con datos de entrada d-dimensional, con L capas ocultas, N(k) unidades en el k-ésimo nivel y una salida de k-dimensional. Donde k es la salida de la capa de la k-ésima, por lo tanto:

$$y_j^{(k)} = \varphi(z_j^{(k)}) \tag{5.26}$$

Todas las capas ocultas con la misma función de activación $\varphi()$ donde $(1 \leq k \leq L+1)$. Los experimentos se han realizado con la siguiente configuración de parámetros [202]:

- Función de activación: Las neuronas de las capas ocultas utilizan una función no lineal de activación. Las tres funciones de activación que se utilizaron en esta investigación fueron: Tanh (Tangente hiperbólica), Rectificador (Elige el máximo valor de entrada) y Maxout (Elige el máximo de las coordenadas del vector).

- Capas ocultas: Este parámetro indica el número y el tamaño de cada capa oculta en el modelo. Por ejemplo: (10,5,10) significa que el modelo tiene tres capas ocultas, donde la primera capa tiene 10 neuronas, la segunda tiene 5 neuronas y la tercera tiene 10 neuronas.

- Épocas: Se refiere al número de pasadas sobre la base de datos para construir el modelo.

- Ratio de aprendizaje: El aprendizaje hace una serie de cálculos de operaciones y mide la derivada de cada paso.

- Épsilon: El épsilon es similar al ratio de aprendizaje durante el aprendizaje inicial y el impulso en las etapas posteriores que permita avanzar hacia adelante.

- Ratio de *annealing*: Esta tasa sirve para reducir el nivel de convergencia a mínimos locales.

- Equilibrio entre clases: Para los datos que tienen más muestras de una clase que de otra, este parámetro permite entrenar modelos con proporciones similares, lo que puede generar modelos más precisos.

5.3.3. Configuración del método RFE

Se aplica el método *wrapper* RFE mediante la técnica *Repeated Cross-validation* (CV)[91] con cinco repeticiones y diez particiones. Lo cual crea 50 modelos. Para los experimentos en este capítulo se utiliza el método *Recursive Feature Elimination* (RFE) [118]. RFE es un método de tipo *wrapper*, es decir, utiliza la precisión predictiva de un determinado algoritmo de aprendizaje para determinar la calidad de las características seleccionadas.

Su entrada es una combinación de los predictores y su salida es un valor que se utiliza como una medida apropiada para evaluar la exactitud del modelo. Para medir la precisión de los modelos se ha utilizado el algoritmo *Random Forest* [203].

Los resultados obtenidos con los métodos *wrapper* suelen ser mejores porque tienen la capacidad de detectar variables correlacionadas o redundantes. Los métodos DL funcionan mejor con todas las variables, ya que estos métodos tienen la capacidad de encontrar relaciones que el resto de métodos no descubre gracias a sus capas ocultas que incrementan la capacidad de abstracción. Por lo tanto, es mejor no aplicar el método de selección de variables RFE para DL.

Para seleccionar el mejor método y la mejor configuración para cada método se aplica el Algoritmo 5.1 que ya está implementado en el paquete Caret. Cada método supervisado tiene varios parámetros que pueden configurarse. Caret prueba varias combinaciones de parámetros automáticamente. Se aplica CV cinco veces con diez particiones para evaluar todos los modelos generados. La exactitud de la configuración del modelo viene dada por el promedio de la exactitud de todos estos modelos.

5.3.4. Base de datos para los experimentos

5.3.4.1. Base de datos para en la detección de *Spam*

Para construir este modelo de clasificación se utilizó un conjunto de datos que consta de 3.279 anuncios, de los cuales 2.821 anuncios son *spam* y 458 no lo son [204]. Cada una de las muestras contiene 1.558 atributos. El 28 % de los casos contiene ruido o falta de valores. Estos valores fueron tratados con la función *"ReplaceMissingValues"* del Software WEKA.

Después de aplicar la función *"NearZeroVar"* la base de datos se redujo de 1.558 atributos a 21. A continuación, se aplicó el método RFE y la base de datos se redujo a solamente 6 atributos.

La principal ventaja de este método es su gran capacidad de abstracción a través de varios niveles jerárquicos, por lo que utiliza la base de datos original sin aplicar selección de variables para explorar su potencial. Para evaluar los métodos de detección de correo no deseado, se elige la métrica RMSE porque se necesita conocer la probabilidad de que un anuncio sea de tipo *spam*. De esta manera, si un anuncio tiene una alta probabilidad de ser *spam*, probablemente no se mostrará. Para todos los modelos se ha seguido la metodología descrita en la Figura 5.2.

5.3.4.2. Base de datos en el modelo CTR

La base de datos utilizada para construir este modelo de clasificación está compuesto de 45.840.617 muestras [205]. La base de datos puede verse como una tabla donde cada fila corresponde a una visita del usuario y cada columna representa una característica del usuario o de una página web donde se muestra el anuncio. La primera columna representa la salida del modelo y tiene dos posibles valores $(0, 1)$, para indicar si el usuario hace clic en el anuncio.

La base de datos tiene 13 columnas con valores numéricos y 26 valores nominales que corresponden a determinadas categorías. Los valores de la categoría fueron codificados usando un *hash* de 32 bits para asegurar la privacidad. Algunos de los campos más importantes son la hora en que el usuario accede la página, la posición del *banner*, el nombre de la página web, el nombre del dominio, la categoría a la que pertenece la página, el tipo de dispositivo, la aplicación desde la cual el usuario accede, el modelo del dispositivo y el tipo de conexión. En primer lugar se aplica la función de *"NearZeroVar"* y así se reduce el número de columnas de 40 a 39. La salida del modelo es la probabilidad de que un usuario haga clic en el *banner* de una página web.

El CTR es un valor de probabilístico, y por lo tanto está entre 0 y 1. Por ejemplo, si dos anuncios tienen los mismos valores para todos los parámetros excepto para el CTR,

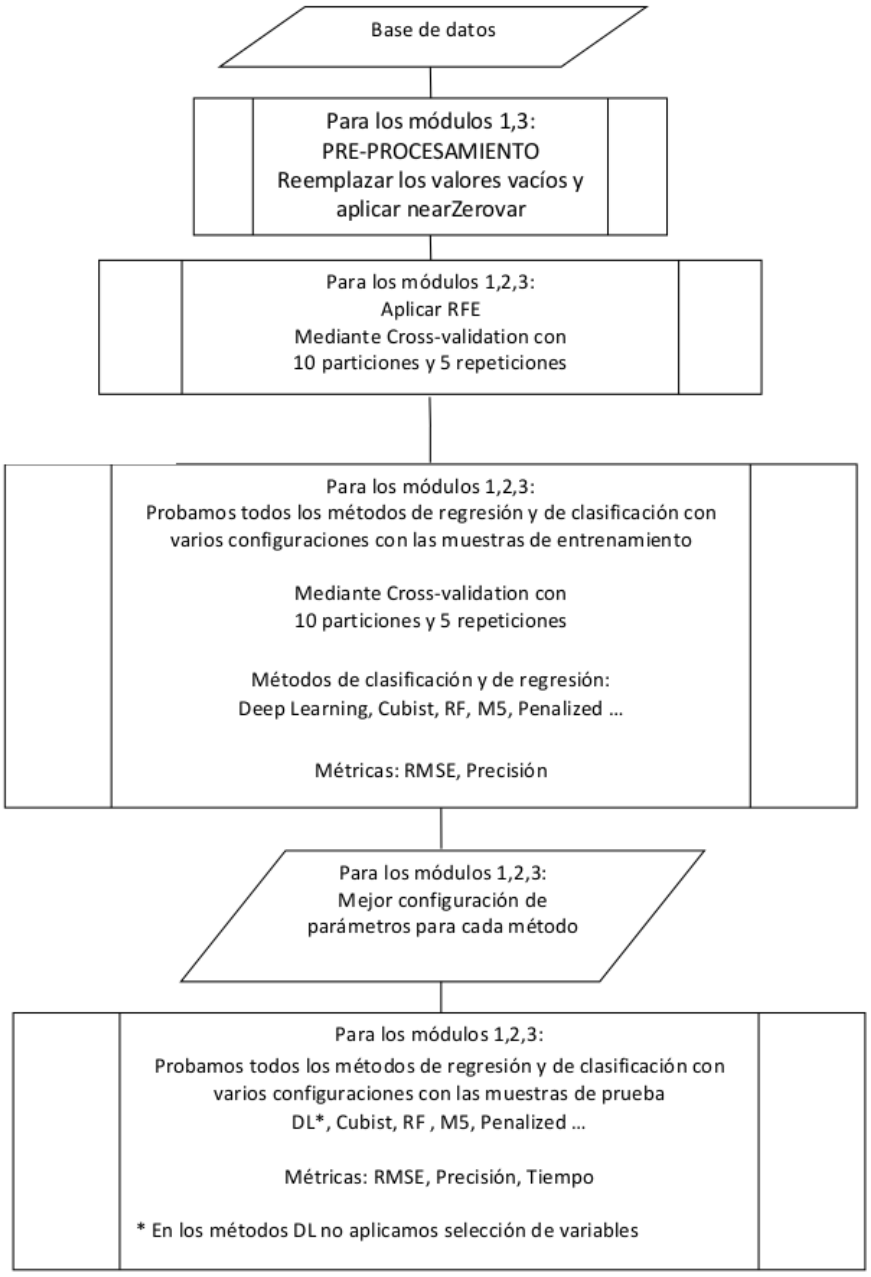

Figura 5.2: Metodología para la construcción de modelos supervisados.

luego el CTR ayudará a seleccionar el mejor anuncio para mostrar. Para la evaluación de los modelos CTR se ha elegido la métrica RMSE. Esta medida es muy apropiada porque ayuda a seleccionar el modelo que tiene un mínimo error en la predicción de la CTR.

5.3.4.3. Base de datos en la predicción de ventas

Este modelo estima el número de veces que un producto se venderá a través de Internet para los próximos 12 meses [206]. Esta estimación permite modificar algunas características del producto para lograr mayores ventas, para cancelar productos no rentables antes de ponerlos en el mercado o para lanzar campañas de publicidad enfocadas a promover los productos más rentables.

En la publicidad *online*, casi ninguna empresa pone a disposición sus datos de ventas debido a que podría dar una ventaja a sus empresas rivales o comprometer los datos del cliente. Por esta razón, es difícil obtener bases de datos de ventas reales. Por lo tanto, se ha utilizado los datos de la comunidad de datos científicos: `Kaggle.com/c/online-sales`.

Kaggle es una plataforma para competiciones de modelos predictivos que tiene expertos de más de 100 países y de 200 universidades que abarcan muchos campos e industrias de estadística, econometría, matemáticas, física, etc. Recientemente, Kaggle ha dejado sus bases de datos disponibles para el uso académico. Estos datos son de la competición de ventas *online* de un producto. La competición consiste en predecir el número de ventas *online* de un producto basado en algunas características de los productos. Esta base de datos contiene 751 instancias que representan diferentes productos.

Para resolver este problema se transforma la base de datos en 12 bases de datos, uno por cada mes del año. Cada base de datos contiene 751 casos, 546 atributos de entrada y un atributo de salida que representa el número de ventas *online* para ese mes.

Los primeros 12 columnas en esta base de datos (resultado M1 a M12 resultado) contiene las ventas para los primeros 12 meses después del lanzamiento del producto. Las 546 columnas restantes son los atributos de entrada.

Date_1 representa el número de días de la campaña de publicidad posteriores al lanzamiento del producto. *Date_2* es el número de días de la campaña publicitaria previos al lanzamiento del producto. Otras columnas de la base de datos representan características del producto y la campaña publicitaria.

Cuan X son variables cuantitativas y *Cat* X son variables categóricas. Las variables categóricas binarias se miden como “1” si el producto tiene la característica y “0” en caso contrario. Las bases de datos contienen valores vacíos en los atributos de entrada y de salida.

Después de usar la función *"NearZeroVar"*, se redujo el número de variables de 546 a

163. La función *"NearZeroVar"* es de tipo filtro por lo que las variables son analizadas y descartadas de manera independientemente.

Después de descartar estas variables se construyeron 12 tablas. Cada tabla tiene todas las entradas de la base de datos y una salida que se corresponde con uno de los doce meses. El promedio del número de atributos para las 12 tablas se redujo de 163 a 87,5. Para elegir el mejor método se aplicó la metodología descrita en el Algoritmo 5.1.

5.3.5. Resultados y discusión

Para probar este sistema se han utilizado tres conjuntos de datos diferentes y se han evaluado de manera independiente. Sin embargo, para calcular la precisión del sistema se debe evaluar con un único conjunto de datos con un total de N entradas, donde N es un número suficiente de muestras.

Cada entrada debe incorporar todos los parámetros requeridos por los módulos 1, 2 y 3. Estos parámetros están relacionados con la visita del usuario, con la campaña del anunciante y con el sitio web del anunciante. Además, es necesario tener un campo que indique si el usuario hace clic, un campo que indique si la compra se ha realizado y un campo que indique si un anuncio es spam.

Una vez que se tenga el conjunto de datos con toda esta información se debería establecer una métrica para medir el desempeño de la metodología. Una métrica adecuada podrían ser los ingresos recaudados por la red, teniendo en cuenta que se puede penalizar económicamente para cada anuncio de tipo spam sobre el que se hizo clic.

No se ha encontrado una base de datos que cumpla con estas características, pues pocas plataformas publicitarias aplican CPM, CPC y CPA al mismo tiempo. Además, cada plataforma selecciona características diferentes de los usuarios y de las páginas. Por lo general, la información correspondiente a los anunciantes y los editores se guarda como confidencial, de ahí que cuando las redes hacen pública esta información se codifican los datos.

Finalmente, el objetivo de esta investigación no es obtener un sistema totalmente desarrollado, sino una metodología general que puede ser adaptada por cualquier red publicitaria.

* Configuración 1: Tres capas ocultas con 500 nodos, 100 épocas, 0.01 de ratio de aprendizaje, 0.001 de *annealing rate*, clases no balanceadas y función de activación “Rectifier”.

** Configuración 2: Tres capas ocultas con 250 nodos, 100 épocas, 0.01 de ratio de aprendizaje, 0,001 de *annealing rate*, clases balanceadas y función de activación: "RectifierWithDropout".

Nº	Nombre del Método	Mejor ROC	ROC Comp.	ROC RFE	Prec. Comp.	Prec. RFE	Tiempo(Seg)
1	Deep Neural Networks	0,9713	0,9713	-	0,9848	-	8513,52
2	C5.0	0,9606	0,9606	0,7840	0,9776	0,9613	31,52
3	Stochastic Gradient Boosting	0,9532	0,9532	0,9012	0,9744	0,9588	6,31
4	Boosted Classification Trees	0,9466	0,9466	0,9056	0,9655	0,9505	77,88
5	Boosted Logistic Regression	0,9265	0,9265	0,8531	0,9651	0,9495	5,14
6	SVM with Radial Basis Function Kernel	0,9174	0,9174	0,8235	0,9674	0,9508	15,71

Tabla 5.1: RMSE de los modelos de clasificación para la detección de *spam*.

Nº	Nombre del Método	Mejor ROC	ROC Comp.	ROC RFE	Prec. Comp.	Prec. RFE	Tiempo(Seg)
1	Deep Neural Networks	0,7722	0,7722	-	0,6990	-	281,56
2	Stochastic Gradient Boosting	0,7160	0,7153	0,7160	0,6578	0,6575	6,31
3	Boosted Classification Trees	0,7117	0,7117	0,7111	0,6526	0,6544	77,88
4	C5.0	0,7057	0,7057	0,7057	0,6517	0,6529	31,52
5	SVM with Radial Basis Function Kernel	0,6902	0,6872	0,6902	0,6365	0,6362	15,71
6	Boosted Logistic Regression	0,5857	0,5857	0,5583	0,6105	0,6107	5,14

Tabla 5.2: RMSE de los modelos de clasificación para la estimación del CTR.

5.3.5.1. Resultados del modelo para la detección de *spam*

Como se muestra en la Tabla 5.1, DL es el algoritmo que mejor predice si un anuncio spam. Aunque la diferencia entre DL y los métodos C5.0 y *Stochastic Gradient Boosting* no parece muy significativa. En la Tabla 5.1, la métrica más importante es el ROC, ya que este modelo se utiliza para indicar la probabilidad de que un anuncio sea spam.

La salida estará en el rango [0, 1] y se utilizará en la fórmula para calcular el valor de un anuncio. Aunque el tiempo de construcción del modelo DL es superior al resto de los modelos, esto generalmente no es una gran desventaja pues el modelo no debe ser creado en tiempo real. El factor crítico es el tiempo de predicción del modelo y DL es similar al resto de los modelos.

5.3.5.2. Resultados del modelo de predicción del CTR

En la estimación del CTR, el ganador es claramente DL. Como se muestra en la Tabla 5.2, la diferencia entre DL y el segundo y tercer mejor modelo es muy significativa.

En la Tabla 5.3 se muestran las mejores configuraciones de los métodos para la detección de spam y para la estimación del CTR.

Nº	Método	Conf.	Parámetros	Valores Spam	Valor CTR
1	Boosted Classification Trees	9	iter, maxdepth, nu	150, 3, 0,1	150, 3, 0,1
2	Boosted Logistic Regression	3	nIter	31	21
3	C5.0	12	trials, model, winnow	20, 2, FALSE	20, 1, FALSE
4	Deep Neural Networks	3	-	Conf. 1*	Conf. 2**
5	Stochastic Gradient Boosting	9	n.trees, interaction.depth, shrinkage	150, 3, 0,1	150, 3, 0,1
6	SVM with Radial Basis Function Kernel	3	sigma, C	0,1071, 1	0,0194, 0,5

Tabla 5.3: Configuración de métodos en la detección de Spam y estimación del CTR.

Nº	Método	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1	Cubist	26703,6	13128,6	6661,5	4142,6	3384,5	1698,5	1380,0	1375,1	1348,7	1082,9	792,8	865,5
2	Deep Learning	25801,4	14578,4	6711,3	4225,7	4145,7	1837,7	1449,8	1411,0	1302,3	1262,7	788,4	953,7
3	Linear Regression	37214,8	17732,9	8428,8	5259,5	5536,6	2229,6	1551,8	1587,9	1612,3	1324,6	983,7	1047,3
4	Model Rules	33975,0	18341,4	8417,6	5224,1	4963,6	2286,7	1568,0	1537,5	1576,0	1257,7	915,2	978,4
5	Model Tree	32073,8	16458,9	8115,0	4490,1	4750,1	1872,3	1538,2	1549,7	1450,7	1247,1	841,6	1002,3
6	Random Forest	29873,8	14535,9	6834,7	4298,0	3763,4	1905,3	1343,8	1293,3	1285,3	1078,2	783,5	851,8

Tabla 5.4: RMSE predicción de ventas utilizando el dataset completo de datos.

Nº	Método	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1	Cubist	26177,8	13077,0	6502,4	4087,7	3537,4	1656,2	1343,0	1376,3	1269,2	1104,7	808,2	861,9
2	Deep Learning	-	-	-	-	-	-	-	-	-	-	-	-
3	Linear Regression	35603,0	17447,0	8237,0	4954,7	4954,4	2152,8	1472,7	1534,7	1574,3	1320,9	895,9	1036,3
4	Model Rules	33304,2	16917,4	8362,8	4633,9	4613,9	2124,5	1511,8	1490,7	1615,0	1242,4	881,2	982,2
5	Model Tree	30466,7	14957,8	8225,9	4302,2	4217,5	1933,1	1456,5	1450,4	1441,0	1232,9	854,6	953,1
6	Random Forest	29247,1	14295,7	6861,7	4168,7	3543,0	1797,9	1275,0	1282,4	1241,7	1076,0	755,8	855,3

Tabla 5.5: RMSE predicción de ventas utilizando el dataset RFE seleccionado.

5.3.5.3. Resultados del modelo de predicción de ventas

En los pronósticos de ventas del experimento, se utilizaron cinco algoritmos de regresión y luego se comparó su rendimiento con DL. En primer lugar, se construyen modelos utilizando la base de datos completa filtrada por la función de "NearZeroVar", como se muestra en la Tabla 5.4. Posteriormente, se utilizó el conjunto de datos seleccionado por el método RFE y los resultados se muestran en la Tabla 5.5.

Por último, como se muestra en la Tabla 5.6, se ha seleccionado el mínimo RMSE para cada mes entre el conjunto completo de datos y el dataset RFE para tasa de modelos. Posteriormente, se ha dividido el valor de cada mes entre el máximo RMSE de cada mes. Finalmente, se calculó el promedio de los doce meses y se clasificaron los métodos de mejor a peor. *Cubist* y *Random Forest* son superiores para a *Deep Learning*.

En la Tabla 5.7 se muestra la configuración de los parámetros para los métodos de predicción de ventas.

*Configuración 3: La configuración empleada en las DNNs para la predicción de ventas es la siguiente: función de activación "RectifierWithDropout", 100 épocas y tres capas ocultas de 250 nodos. El resto de los parámetros fueron configurados con los valores predeterminados.

Nº	Método	Prom	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1	Cubist	0,806	0,735	0,750	0,778	0,825	0,683	0,769	0,888	0,896	0,805	0,820	0,885	0,832
2	Random Forest	0,816	0,821	0,819	0,817	0,841	0,715	0,835	0,843	0,836	0,788	0,815	0,844	0,822
3	Deep Learning	0,864	0,725	0,836	0,803	0,853	0,837	0,854	0,959	0,919	0,826	0,956	0,880	0,920
4	Model Tree	0,907	0,856	0,857	0,970	0,868	0,851	0,870	0,963	0,945	0,914	0,933	0,939	0,920
5	Model Rules	0,967	0,935	0,970	1,000	0,935	0,931	0,987	1,000	0,971	1,000	0,941	0,984	0,944
6	Linear Regression	0,997	1,000	1,000	0,985	1,000	1,000	1,000	0,974	1,000	0,999	1,000	1,000	1,000

Tabla 5.6: Clasificación de los métodos de regresión de predicción de ventas.

Nº	Método	Tiempo(Seg)	Conf.	Parámetros	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1	Cubist	1,09	9	committees, neighbors	20, 9	20, 9	20, 9	20, 9	20, 5	20, 9	20, 9	20, 9	20, 9	20, 9	20, 9	20, 5
2	Deep Learning	170,52	1	*Conf 3	-	-	-	-	-	-	-	-	-	-	-	-
3	Linear Regression	0,11	1	parameter	1	1	1	1	1	1	1	1	1	1	1	1
4	Model Rules	2,51	4	pruned, smoothed	1, 1	1, 1	1, 1	1, 1	1, 2	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1
5	Model Tree	1,62	8	pruned, smoothed, rules	2, 1, 2	1, 1, 2	1, 1, 1	1, 1, 2	2, 2, 2	1, 1, 2	2, 1, 2	1, 1, 2	1, 1, 2	1, 1, 2	2, 1, 2	1, 1, 2
6	Random Forest	10,89	3	mtry	82	163	82	82	82	82	82	82	82	82	82	82

Tabla 5.7: Configuración de parámetros de los modelos de predicción de ventas.

5.4. Conclusiones

En este capítulo se propone una metodología para el cálculo del valor de un anuncio en CPM, CPC y CPA mediante preprocesamiento, selección de características y la construcción de métodos supervisados de clasificación y de regresión. Para la selección de características se han utilizado los métodos RFE y DLNNs. Los métodos supervisados se han evaluado mediante las métricas ACC y ROC en clasificación, y mediante la métrica RMSE para regresión. Los resultados se han comparado con cinco métodos muy conocidos de clasificación y de regresión.

Después de trabajar de manera más de cercana con DL, se puede concluir que tiene una serie de ventajas y una serie de desventajas. Las principales ventajas de DL son: a) El rendimiento de DL para problemas de clasificación es significativamente mayor que el resto de los métodos. b) DL ahorra el proceso de selección de variables. Lo cual, requiere mucho tiempo, especialmente en los métodos *wrapper*. c) DL puede aplicarse a muchos dominios como son la detección de spam, la estimación CTR y la predicción de ventas.

Las principales desventajas de DL son: a) Es computacionalmente muy costoso. b) No se ha definido una metodología para una configuración de parámetros adecuada. c) La complejidad de las capas ocultas de DL dificulta la interpretación de los resultados y la comprensión del algoritmo.

Las DLNNs obtuvieron el primer lugar en la detección de spam y en la estimación del CTR. Pero su desempeño en la predicción de ventas quedó por debajo de las expectativas. El buen rendimiento de los métodos DL probablemente se deba al alto nivel de abstracción modelado por estos métodos. Debido a que utilizan arquitecturas de múltiples capas encadenadas basadas en transformaciones no lineales.

Los métodos DL modelan abstracciones de alto nivel usando múltiples capas y extracción jerárquica de entidades. Además, los métodos DL utilizan la extracción de variables para limpiar el ruido en los datos. Los modelos DL no están diseñados para resolver un único tipo de problema sino que pretenden abarcar todo tipo de problemas, especialmente aquellos relacionados con el reconocimiento de imágenes y de sonido. Esto nos permite vislumbrar el gran futuro que tiene DL, no sólo en problemas relacionados con la publicidad en Internet sino también en la predicción de ventas.

La principal ventaja de esta metodología es que es muy fácil de implementar pues existen herramientas ML que generan automáticamente el modelo. Además, estos métodos son bien conocidos y se han implementado con éxito en varias empresas internacionales [207].

Una desventaja de DL en la publicidad en internet es que los modelos no pueden ser interpretados. Esto significa que es posible que no se pueda saber por qué un anuncio está clasificado como *spam* o por qué el CTR de un anuncio es bajo. Esta información podría ser útil si se quisieran analizar los resultados y averiguar las áreas de mejora para este sistema.

Una interesante línea de mejora para obtener un mejor rendimiento sería hacer múltiples réplicas del modelo para que se pueda equilibrar la carga de trabajo y por lo tanto, reducir el tiempo de respuesta.

El sistema desarrollado podría ser mejorado agregando mecanismos para detectar fraude en la publicidad en línea. Esto podría hacerse añadiendo tres módulos que se pueden llamar A, B y C. El módulo A se utilizará en las redes CPM y medirá la probabilidad de que el editor esté realizando impresiones publicitarias fraudulentas. El módulo B se empleará en las redes de CPC para evaluar la probabilidad de que un clic sea fraudulento. Y el módulo C se utilizaría en las redes CPA para calcular la probabilidad de que una acción sea fraudulenta. Un ejemplo de acción fraudulenta podría ser que un editor le diga a un amigo que rellene un formulario con la única intención de obtener una comisión.

Este capítulo está centrado en calcular el valor en dólares de un anuncio, pero para decidir qué anuncio mostrar, es necesario tener en cuenta otros factores. Una interesante línea de investigación sería desarrollar una metodología mejorada para calcular el valor del parámetro $Ranking\ Anuncio$. El nivel de calidad debe tener en cuenta muchos factores, como si todos los anunciantes han podido mostrar los anuncios, si el precio que los anunciantes pagan por los anuncios es similar al precio de mercado, el nivel de satisfacción de los usuarios cuando visitan el sitio web del anunciante, el CTR del anuncio a lo largo de la campaña y así sucesivamente. Una adecuada estimación del $Ranking\ Anuncio$ permitirá decidir adecuadamente sobre el mejor anuncio entre todos aquellos que luchan por aparecer en la visita de un usuario.

Capítulo 6

Optimización de los módulos del valor de un anuncio mediante ENORA y otros métodos de selección de variables

6.1. Introducción

En el presente capítulo se aplican varios métodos de selección de variables para mejorar el rendimiento del módulo de predicción de ventas (regresión) y para mejorar el método de estimación del CTR (clasificación).

El algoritmo NSGA-II es sin duda el algoritmo más utilizado y una de las estrategias de búsqueda de referencia. Este capítulo se centra en el problema de aprendizaje supervisado de regresión y en la estimación del CTR. Para lograr simultáneamente una precisión alta y un número reducido de características se utilizan los métodos *wrapper* de selección de subconjuntos con estrategias de búsqueda evolutiva multiobjetivo. También se utiliza *Random Forest* como método de aprendizaje de regresión porque corrige la tendencia de los árboles de decisión al sobreajuste, que es un ajuste excesivo a los datos de entrenamiento, en su sistema del entrenamiento, y que puede ser utilizado para clasificar la importancia de las variables, y posteriormente se puede utilizar como métrica de la estructura interna de los datos [208, 209].

Además, *Random Forest* funciona de manera eficiente en grandes bases de datos. Esta propuesta también se compara con otro método de selección de variables de tipo *wrapper* bastante conocido llamado *Recursive Feature Elimination* (RFE). Con el fin de predecir las ventas *online* de un producto se emplea una base de datos con las características del producto. Una correcta estimación del número de ventas es muy útil para lanzar campañas publicitarias de los productos más rentables. También puede servir para modificar algunas

características del producto hasta encontrar la combinación que genere el modelo que prediga un mayor número de ventas.

Predecir con exactitud el número de ventas que se producirá en el futuro puede ser decisivo para el éxito de un negocio. Imagine el potencial que puede tener para cualquier empresa conocer el número de ventas que se producirá al siguiente año. Si las ventas son inferiores a lo esperado, es posible que la compañía se vaya a la quiebra porque no puede pagar a los proveedores, y si los productos solicitados son mayores que el número de unidades disponibles, la compañía habrá perdido una oportunidad de crecimiento.

Lo mismo sucede con el rendimiento del CTR, basta imaginar las ganancias que se obtendrían si se lograra estimar de manera precisa la probabilidad de que se haga clic en un anuncio. Si se tiene en cuenta que solamente en una hora en internet se producen miles de millones de impresiones de anuncios, elegir siempre el mejor anuncio supone millones de dólares anuales.

Para crear modelos precisos, es necesario seleccionar las variables adecuadas y encontrar los métodos adecuados para el problema que se esté tratando. Con el fin de eliminar características se propone un método de selección de tipo *wrapper* donde el método *Random Forest* es utilizado para evaluar los modelos de regresión. El algoritmo *Random Forest* funciona mediante la construcción de muchos árboles de decisión durante la creación del modelo y la salida es el promedio de la previsión de los árboles individuales.

En este capítulo se propone un algoritmo evolutivo multiobjetivo llamado ENORA [152, 153]. Esta estrategia de búsqueda se compara para regresión (predicción de ventas) con el famoso algoritmo evolutivo multiobjetivo NSGA-II tanto desde el punto de vista de los resultados estadísticos como en la interpretación del proceso de búsqueda [137]. El algoritmo NSGA-II es la referencia en la mayoría de los estudios realizados. También se compara ENORA con el método recursivo de eliminación de variables *Recursive Feature Elimination* (RFE).

En este capítulo también se aplica la selección de variables para clasificación mediante varios métodos de selección de variables para la estimación del CTR. En concreto, se aplican los métodos *Gain Ratio*, PCA, RFE, ENORA y NSGA-II. En el modelo CTR se aplican 12 métodos de selección de variables para ver cuál es el método de selección de variables más adecuado para estimar el CTR.

El resto de este capítulo tiene la siguiente estructura: En la sección 2, se describen las principales cuestiones preliminares. En la sección 3, se describe en detalle el algoritmo evolutivo multiobjetivo ENORA para selección de variables en los métodos de regresión. La sección 4 muestra los resultados de los experimentos utilizando las bases de datos de las ventas *online* y el modelo CTR. Se propone una metodología que incluye preprocesamiento

de los datos para la selección de variables multiobjetivo, la comparación de optimizadores de rendimiento, el proceso de decisión y para el análisis y construcción de los modelos de regresión. Por último, en la última sección se discuten las novedades y las ventajas de esta metodología. También se exponen las conclusiones del documento y los trabajos futuros.

6.2. Selección de las características mediante ENORA

En esta sección se describen los componentes principales de ENORA adaptado a la selección de variables. Estos componentes son la representación de la solución, la población inicial, la selección y toma de muestras, los esquemas de reemplazo generacional y la variación.

6.2.1. Representación de soluciones y evaluación

Cada individuo se representa mediante un conjunto de bits de longitud fija. Cada bit representa un atributo de la base de datos. Si un bit es "1", significa que este atributo está seleccionado para el subconjunto reducido y si es "0" indica que el atributo no está seleccionado.

Por lo tanto, la longitud de los individuos es igual al número de atributos M en el conjunto inicial de datos. Además, para llevar a cabo un cruce y mutación autoadaptativa cada individuo tiene dos parámetros discretos $d_I \in \{0, \ldots, \delta\}$ y $e_I \in \{0, \ldots, \epsilon\}$ asociados al cruce y a la mutación, donde $\epsilon \geq 0$ es el número de operadores de cruce y $\delta \geq 0$ es el número de operadores de mutación. Por lo tanto, cada individuo I se representa como:

$$I = \left\{b_1^I, \ldots, b_M^I, d_I, e_I\right\},\ b_i^i \in \{0,1\},\ i = 1, \ldots, M,\ d_I \in \{0, \ldots, \delta\},\ e_I \in \{0, \ldots, \epsilon\}$$

Un individuo I se evalúa con dos funciones de aptitud, $f_1(I)$ y $f_2(I)$, que se corresponden con los dos objetivos a minimizar del modelo de optimización multiobjetivo:

$$f_1(I) = RMSE(I)$$
$$f_2(I) = C(I)$$

Se utiliza RMSE pues es la métrica más frecuente para evaluar los métodos de regresión. $\mathcal{C}(I)$ es la cardinalidad del subconjunto que representa al individuo I, por ejemplo el número de bits iguales a 1 en el individuo I.

6.2.2. Población inicial

La población inicial se genera aleatoriamente como se describe en el Algoritmo 6.1. Para cada individuo de la población se genera inicialmente un número aleatorio $q \in \{1, \ldots, N\}$. Posteriormente, se fijan a 1 q bits al azar en el individuo I y los restantes $N - q$ bits se fijan a 0. Por último, se generan aleatoriamente los valores d_I y e_I para la variación autoadaptativa desde $\{0, \delta\}$ hasta $\{0, \epsilon\}$ respectivamente.

Algoritmo 6.1 Iniciar población.

Entrada: $\delta > 0$ ▷ Número de operadores de cruce
Entrada: $\epsilon > 0$ ▷ Número de operadores para la mutación
Entrada: N ▷ Número de atributos de entrada en la base de datos
Entrada: $popsize \geq 0$ ▷ Número de individuos en la población

1: $P \leftarrow$ Vaciar población
2: **for** $I = 1$ a $popsize$ **do**
3: $I \leftarrow$ Nuevo individuo
4: $Q \leftarrow \{1, \ldots, N\}$
5: $q \leftarrow$ Int *Aleatorios* de Q
6: $r \leftarrow N - q$
7: **for** $i = 1$ a q **do**
8: $j \leftarrow$ *Aleatorios* de Q
9: $b_j^I \leftarrow 1$
10: $Q \leftarrow Q - \{j\}$
11: **end for**
12: **for** $i = 1$ to r **do**
13: $j \leftarrow$ *Aleatorios* de Q
14: $b_j^I \leftarrow 0$
15: $Q \leftarrow Q - \{j\}$
16: **end for** ▷ Discretos aleatorios para variación autoadaptativa
17: $d_I \leftarrow$ Int *Aleatorio* de $\{0, \delta\}$
18: $e_I \leftarrow$ Int *Aleatorio* de $\{0, \epsilon\}$
19: Añadir I a la población P
20: **end forreturn** P

6.2.3. Operadores de variación

La mutación y el cruce autoadaptativos se utilizan para cumplir los objetivos de mantenimiento de una diversidad en la población y de asegurar la convergencia de los algoritmos evolutivos. En un algoritmo evolutivo autoadaptativo, las probabilidades de cruce y mutación varían dependiendo de las soluciones del valor de la función de adaptación [210].

Al usar operadores de variación autoadaptativos como el Algoritmo 6.4, no es necesario establecer *a priori* la probabilidad de aplicación de los distintos operadores. En cada etapa

se utiliza el operador de cruce uniforme y operador de mutación, aunque se puede considerar cualquier otro tipo de operador de variación.

La selección de los operadores se realiza por medio de la técnica de adaptación que utiliza los parámetros d_i y e_i para indicar que cruce y que mutación se lleva a cabo por el individuo I. Esta técnica de cruce y mutación aparece en los Algoritmos 6.2 y 6.3 respectivamente. El Algoritmo 6.4 se utiliza para generar dos hijos a partir de dos padres por mutación y cruce autoadaptativo. El cruce uniforme evalúa cada bit en el cromosoma del padre para hacer intercambios con una probabilidad de 0,5. La mutación se hace en una ejecución y afecta a un bit del padre.

Algoritmo 6.2 Cruce adaptativo.

Entrada: I, J ▷ Individuos para el cruce
Entrada: p_v $(0 < p_v < 1)$ ▷ Probabilidad de un cambio de operador
Entrada: $\delta > 0$ ▷ Número de operadores de cruce diferentes($\delta = 1$ en este caso)

1: **if** Una variable aleatoria de tipo Bernoulli con probabilidad p_v toma el valor 1 **then**
2: $d_I \leftarrow$ Int *Aleatorio* desde $\{0, \delta\}$
3: **end if**
4: $d_J \leftarrow d_I$
5: Procesar el tipo de cruce especificado por d_I: ▷ 0: No hay cruce ▷ 1: Cruce uniforme

Algoritmo 6.3 Mutación adaptativa.

Entrada: I ▷ Individuos para mutar
Entrada: p_v $(0 < p_v < 1)$ ▷ Probabilidad de un cambio de operador
Entrada: $\epsilon > 0$ ▷ Número de diferentes operadores de mutaciones ($\epsilon = 1$ en este caso)

1: **if** Una variable aleatoria de Bernoulli de probabilidad p_v toma el valor 1 **then**
2: $e_I \leftarrow$ Int *Aleatorios* de $\{0, \epsilon\}$
3: **end if**
4: Desarrolla el tipo de mutación especificada por e_I: ▷ 0: No hay mutación ▷ 1: Una mutación en una etapa

Algoritmo 6.4 Variación.

Entrada: $Padre1$, $Padre2$ ▷ Variar a los individuos

1: $Hijo1 \leftarrow Padre1$
2: $Hijo2 \leftarrow Padre2$
3: Cruce autoadaptativo $Hijo1$, $Hijo2$
4: Mutación autoadaptativa $Hijo1$
5: Mutación autoadaptativa $Hijo2$
6: **return** $Hijo1, Hijo2$

6.3. Experimentos y resultados

6.3.1. Experimento I: Selección de variables en la predicción de ventas

En esta sección se presentan los resultados del experimento utilizando las bases de datos obtenidas mediante una metodología que se aplica por separado a cada uno de las doce bases de datos como se muestra en la Figura 6.1. La metodología propuesta incluye para cada base de datos: el procesamiento de los datos, la selección de variables, la comparación de la optimización del rendimiento (basado en métricas de hipervolúmenes), la construcción del modelo de regresión y las pruebas. Por último, el modelo de regresión resultante se construye con la selección de los mejores modelos de regresión obtenidos para cada mes.

6.3.1.1. Preprocesamiento de datos

A continuación se describe el procedimiento aplicado a la base de datos. En primer lugar, todos los atributos y valores faltantes se sustituyen con la moda y el promedio; para ello, se utiliza la función *"ReplaceMissingValues"* del paquete *weka.filters.unsupervised.attribute*. En segundo lugar, las características que tienen poca o ninguna variación son eliminadas mediante el procedimiento *"NearZeroVar"* del paquete Caret de R Studio[211]. Como resultado, la base de datos original se redujo a 163 características.

6.3.1.2. Selección y toma de decisiones

En este capítulo se utilizan tres métodos de selección de variables diferentes: ENORA, NSGA-II y RFE. ENORA y NSGA-II son métodos probabilísticos de optimización multiobjetivo, y requieren múltiples ejecuciones con diferentes semillas generadas de forma aleatoria. Con ENORA y NSGA-II se realizan los siguientes pasos:

1. Selección de variables: El proceso de selección de variables se realizan con 30 iteraciones con un método *wrapper* basado en el método *Random Forest* y con una estrategia de búsqueda evolutiva multiobjetivo, donde se trata de minimizar el RMSE y también se procura reducir al mínimo el número de atributos seleccionados. La selección de variables con ENORA y NSGA-II se implementa en Java utilizando el paquete Weka. Para cada ejecución de ENORA y NSGA-II se usa el siguiente evaluador:

   weka.attributeSelection.WrapperSubsetEval -B weka.classifiers.trees.RandomForest -F 5 -T 0,01 -R 1 -E acc – -I 10 -K 0 -S 1 -num-slots 1

2. Ambos algoritmos se ejecutaron con un tamaño de población de 1.000 y con 100 generaciones, es decir, se realizaron 100.000 evaluaciones de la función de evaluación con

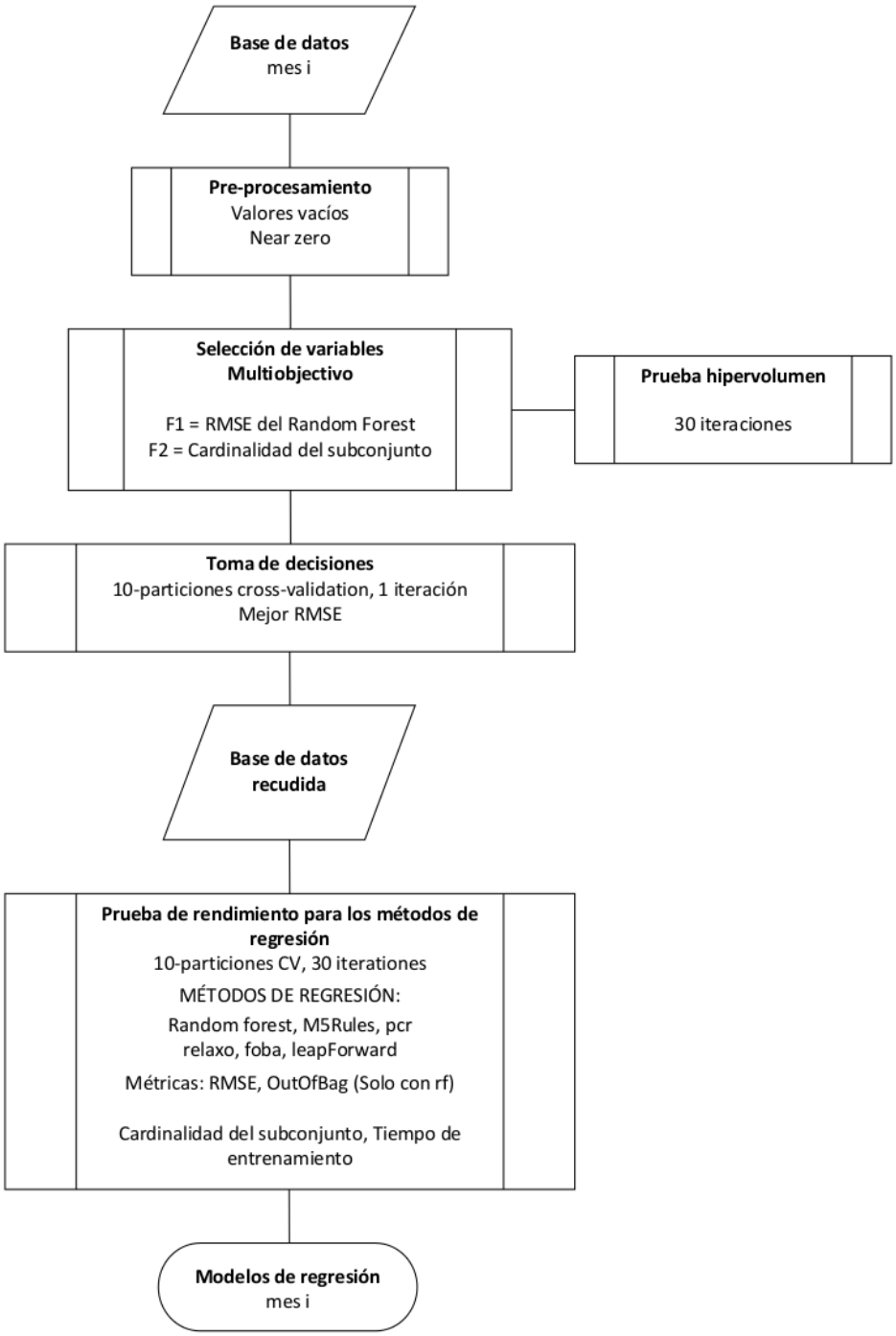

Figura 6.1: Metodología para la construcción de modelos de regresión.

una semilla diferente para cada ejecución en un ordenador con 8 procesadores Intel Xeon X 7550 a 2,00 GHz, RAM 1 TByte a 1067MHz y un almacenamiento de sistema

Mes	ENORA	NSGA-II	RFE
Ene	17	17	95
Feb	15	11	120
Mar	19	11	137
Abr	49	10	78
May	10	13	18
Jun	9	9	16
Jul	9	11	26
Ago	23	14	130
Sep	27	11	80
Oct	14	5	162
Nov	33	9	22
Dic	13	11	157

Tabla 6.1: Atributos seleccionados por ENORA, NSGA-II y RFE.

de archivos distribuido Lustre v2.5.2. y con una red de interconexión Infiniband QDR (40 Gbps). El tiempo para cada ejecución con ENORA ha sido, en promedio, 9,08 h. y con el NSGA-II 5,23 h. ENORA fue incorporado por los autores en Weka como un paquete oficial mediante el paquete *MultiObjectiveEvolutionarySearch* [212].

3. La toma de decisiones para cada serie se realizó con 10 particiones de *cross-validation* en cada solución no dominante y posteriormente se identificó la solución con el mejor valor para el test de *cross-validation*.

4. Construir una base de datos reducidos con los atributos seleccionados.

Para la selección de variables con el método RFE se utiliza el algoritmo descrito en el Algoritmo 2.1. Este algoritmo utiliza selección regresiva, validación de remuestreo y externa para la selección de variables y se implementa con el paquete Caret de R Studio mediante la función *rfe* con el siguiente comando de control:

rfeControl(functions=rfFuncs, method="repeatedcv", repeats=30, number=10)

donde *rfFuncs* define una función de selección llamada *Random Forest*, y *"repeatedcv"* define varias repeticiones de k-particiones de *cross-validation* con 30 ejecuciones de remuestreo y 10 particiones. El algoritmo RFE ha sido ejecutado en una computadora con un procesador Intel (r) Core (TM) i5 2400 CPU a 3,10 GHz con 16Gb, RAM ejecuta Service Pack 1 de Windows 7 Pro 64 bits. El tiempo de ejecución fue 0,84 h. La Tabla 6.1 muestra el número de variables seleccionadas para cada método.

6.3.1.3. Prueba de hipervolúmenes

En esta sección, ENORA y NSGA-II se comparan y se muestran los resultados. El objetivo de este conjunto de experimentos es determinar el algoritmo de optimización con mejor rendimiento.

Para comparar los algoritmos se utiliza el indicador de hipervolúmenes. Este se puede definir en términos generales [128] como el volumen del espacio de búsqueda dominante en una población P, con la ecuación 6.1:

$$HV\left(P\right) = \bigcup_{i=1}^{|Q|} v_i \tag{6.1}$$

donde $Q \subseteq P$ es el conjunto de individuos no dominantes de P y v_i es el volumen del individuo i. Utilizamos, por conveniencia técnica, la relación de hipervolúmenes definida como el cociente entre el volumen del espacio de búsqueda no dominantes sobre el volumen del espacio de toda búsqueda, con la ecuación 6.2:

$$HVR\left(P\right) = 1 - \frac{HV\left(P\right)}{volS} \tag{6.2}$$

donde $volS$ es el volumen del espacio de búsqueda.

Aunque pueden usarse otras métricas de rendimiento de MOEA, se elige la métrica de hipervolúmenes, que mide, simultáneamente, la diversidad y la optimalidad de las soluciones no dominantes. La métrica de hipervolúmenes no requiere el uso de una población óptima, la cual no está disponible para estas pruebas [128].

Otros indicadores como el ratio de error, la distancia generacional, *maximum Pareto-optimal front error, spread, maximum spread, or chi-square-like deviation*, necesitan una población óptima, lo que significa que no es posible calcular estas métricas para las pruebas.

Además, otras métricas como *spacing* sólo miden la uniformidad de las soluciones no dominantes y no toman en cuenta el grado de extensión o la optimalidad. Las Figuras 6.2 y 6.3 muestran la evolución del ratio del promedio de los hipervolúmenes de 30 ejecuciones de ENORA y NSGA-II para las 12 bases de datos.

La Tabla 6.2 muestra los valores estadísticos basados en la métrica de relación de hipervolúmenes de ENORA y de NSGA-II para las 12 bases de datos. La Figura 6.4 muestra los diagramas *boxplots* relacionando estos valores estadísticos.

6.3.1.4. Prueba del rendimiento del modelo de regresión

Después de la selección de variables, el siguiente paso es probar y comparar las bases de datos reducidas utilizando dos tipos de prueba. La primera se desarrolló utilizando *Random*

Mes	Algoritmo	Mínimo	Máximo	Promedio	D.E.	I.C. Bajo	I.C. Alto
Ene	ENORA	0,4707	0,4983	0,4881	0,0061	**0,4859**	**0,4904**
	NSGA-II	0,4822	0,5260	0,5047	0,0101	0,5009	0,5085
Feb	ENORA	0,4791	0,5053	0,4934	0,0070	**0,4907**	**0,4960**
	NSGA-II	0,4847	0,5219	0,4998	0,0091	0,4964	0,5032
Mar	ENORA	0,6492	0,6738	0,6624	0,0062	**0,6601**	**0,6647**
	NSGA-II	0,6616	0,6872	0,6757	0,0071	0,6730	0,6783
Abr	ENORA	0,5869	0,6182	0,6058	0,0066	**0,6034**	**0,6083**
	NSGA-II	0,6015	0,6504	0,6259	0,0111	0,6217	0,6300
Mar	ENORA	0,4091	0,4415	0,4205	0,0063	**0,4181**	**0,4228**
	NSGA-II	0,4138	0,4754	0,4405	0,0171	0,4341	0,4468
Jun	ENORA	0,5768	0,6024	0,5874	0,0061	**0,5851**	**0,5897**
	NSGA-II	0,5823	0,6145	0,6005	0,0088	0,5972	0,6038
Jul	ENORA	0,6777	0,6957	0,6885	0,0046	**0,6868**	**0,6902**
	NSGA-II	0,6835	0,7000	0,6951	0,0044	0,6934	0,6967
Ago	ENORA	0,6653	0,6925	0,6810	0,0069	**0,6784**	**0,6836**
	NSGA-II	0,6736	0,7087	0,6935	0,0107	0,6895	0,6975
Sep	ENORA	0,6091	0,6500	0,6302	0,0109	**0,6261**	**0,6342**
	NSGA-II	0,6271	0,6799	0,6526	0,0139	0,6474	0,6578
Oct	ENORA	0,7078	0,7328	0,7209	0,0055	**0,7188**	**0,7229**
	NSGA-II	0,7218	0,7387	0,7339	0,0053	0,7319	0,7359
Nov	ENORA	0,6500	0,6652	0,6590	0,0040	**0,6575**	**0,6605**
	NSGA-II	0,6534	0,6769	0,6658	0,0055	0,6637	0,6679
Dic	ENORA	0,7288	0,7488	0,7386	0,0050	**0,7367**	**0,7405**
	NSGA-II	0,7338	0,7634	0,7460	0,0072	0,7433	0,7487

D.E. = Desviación estandar del promedio
I.C. = Intervalo de confianza para la media (95 %)

Tabla 6.2: Estadísticas para el ratio del hipervolumen con ENORA y NSGA-II.

Forest como modelo de regresión.

Esta prueba es interesante porque nuestros métodos de selección de variables habían usado *Random Forest* como evaluador del método *wrapper*. La segunda prueba comprueba el funcionamiento de otros modelos de regresión aplicados a la base de datos reducida.

Pruebas con *Random Forest*

Para esta prueba se utiliza la herramienta de minería de datos Weka. Para cada uno de los 12 meses se realizaron dos experimentos:

1. En el primero, la herramienta se configuró usando 4 bases de datos distintas: ENORA, NSGA-II y RFE y la base de datos original y el algoritmo de regresión *Random Forest*, con 50 ejecuciones. Para analizar los resultados de este experimento se realizó *Paired T-Test* corregida con el parámetro de significación 0,05. *Paired T-Test* es una versión corregida de la *Paired T-Test* implementada en Weka para evitar algunos problemas del test original mediante *cross-validation*.

   La base de datos obtenida con ENORA y la métrica *Out of bag* (OOB) fue utilizada como campo de comparación. El objetivo de este experimento fue comparar la métrica OOB obtenida con el algoritmo al Random Forest para los bases de datos considerados. La Tabla 6.3 muestra los resultados de este experimento.

2. En el segundo experimento, la herramienta se configuró con las mismas bases de datos que el primer experimento y el algoritmo de regresión *Random Forest*, con *cross-validation* y 10 particiones. Para analizar los resultados de este experimento se realizó con *Paired T-Test* corregida con el parámetro significación 0,05, las bases de datos obtenidas con ENORA campos de comparación el RMSE, el tamaño del modelo y el tiempo en milisegundos para la construcción del modelo. El objetivo de este experimento es comparar el RMSE, el tiempo de entrenamiento y el tamaño del modelo obtenida con el algoritmo al *Random Forest* para los bases de datos. Las Tablas 6.3, 6.4 y 6.5 muestran los resultados de este experimento.Entre paréntesis la desviación estandar.

   En la Tabla 6.6 la marca $*$ indica que el resultado es estadísticamente peor que el obtenido con ENORA; una marca v indica un resultado estadísticamente mejor, y si no hay marca es que no hay una diferencia estadística significativa. Los valores entre paréntesis son las desviaciones estándar y los resultados en negrita son los mejores.

Mes	ENORA	NSGA-II	RFE	BBDD Original
Ene	**10758, 51(168, 97)**	10774, 97(175, 61)	11618, 91(298, 16) *	12318, 36(303, 28) *
Feb	**5907, 25(103, 84)**	5946, 53(85, 46) *	6463, 95(116, 71) *	6650, 13(169, 22) *
Mar	**3005, 84(37, 75)**	3048, 23(44, 92) *	3273, 69(46, 87) *	3296, 62(42, 63) *
Abr	1833, 55(29, 28)	**1823, 53(27, 22)**	1943, 12(32, 98) *	2023, 21(37, 82) *
May	**1400, 43(22, 87)**	1400, 45(21, 76)	1514, 07(30, 72) *	1709, 91(45, 63) *
Jun	963, 53(8, 86)	983, 55(10, 50) *	**957, 67(12, 88)** v	1062, 39(15, 87) *
Jul	753, 15(7, 54)	**746, 04(8, 26)** v	778, 66(7, 38) *	854, 07(10, 27) *
Ago	**691, 89(9, 18)**	700, 01(7, 43) *	731, 84(9, 34) *	756, 79(10, 09) *
Sep	**639, 79(7, 52)**	683, 75(6, 38) *	656, 14(10, 82) *	707, 42(9, 08) *
Oct	569, 02(6, 42)	**563, 39(7, 26)** v	635, 54(8, 71) *	635, 09(6, 84) *
Nov	486, 61(4, 74)	**473, 01(5, 21)** v	485, 75(6, 17)	550, 82(6, 43) *
Dic	444, 94(5, 16)	**442, 28(5, 87)** v	503, 82(7, 35) *	506, 59(6, 78) *

Tabla 6.3: Medida OOB (50 ejecuciones) pruebas con *Random Forest*.

Mes	ENORA	NSGA-II	RFE	BBDD Original
Ene	**24889, 83(11737, 96)**	25084, 47(11801, 95)	27874, 60(14502, 84)	29461, 60(15623, 85) *
Feb	**11843, 42(3905, 03)**	11923, 77(3932, 47)	14534, 19(5529, 57) *	15002, 61(5845, 31) *
Mar	**6264, 64(1988, 61)**	6368, 28(2113, 90)	6629, 60(2208, 75)	6691, 59(2212, 97)
Abr	**3812, 46(1523, 97)**	3840, 19(1704, 12)	4168, 81(1893, 88)	4315, 07(1954, 86)
May	**2635, 12(957, 58)**	2666, 92(947, 32)	3533, 80(1403, 60) *	3937, 23(1827, 04) *
Jun	**1591, 74(394, 09)**	1615, 20(400, 56)	1750, 28(514, 71)	1846, 14(526, 66) *
Jul	1268, 91(336, 12)	**1258, 10(342, 97)**	1278, 97(349, 15)	1370, 34(386, 83) *
Ago	**1226, 30(382, 12)**	1248, 04(383, 06)	1288, 27(458, 83)	1319, 61(469, 74) *
Sep	**1121, 66(355, 65)**	1195, 19(371, 78) *	1217, 84(461, 51)	1294, 44(503, 85)
Oct	**1004, 53(428, 59)**	1021, 23(431, 14)	1085, 78(499, 24)	1084, 38(499, 42)
Nov	**753, 16(146, 03)**	754, 10(137, 16)	762, 37(143, 18)	826, 44(170, 69) *
Dic	**800, 53(208, 91)**	806, 58(204, 50)	855, 83(244, 04)	857, 44(243, 27)

Tabla 6.4: RMSE (CV 10 part. y 30 ejec.) pruebas con *Random Forest*.

Mes	ENORA	NSGA-II	RFE	BBDD Original
Ene	**118890, 70(51, 99)**	**118889, 70(51, 99)** v	544961, 30(239, 50) *	915365, 90(402, 97) *
Feb	107973, 30(47, 18)	**86132, 50(37, 56)** v	680720, 30(299, 60) *	915365, 90(402, 97) *
Mar	129789, 10(56, 79)	**86126, 50(37, 56)** v	773476, 70(340, 47) *	915365, 90(402, 97) *
Abr	293583, 10(128, 92)	**80648, 30(35, 16)** v	451607, 90(198, 63) *	915345, 90(402, 97) *
May	**80825, 30(35, 16)**	97033, 90(42, 37) *	124207, 90(54, 39) *	915345, 90(402, 97) *
Jun	**75189, 10(32, 75)**	75191, 10(32, 75) *	113289, 50(49, 58) *	915345, 90(402, 97) *
Jul	**75193, 10(32, 75)**	86112, 50(37, 56) *	167851, 50(73, 62) *	915345, 90(402, 97) *
Ago	151617, 90(66, 41)	**102485, 10(44, 77)** v	735283, 30(323, 64) *	915345, 90(402, 97) *
Sep	173459, 70(76, 03)	**86118, 50(37, 56)** v	462477, 30(203, 44) *	915345, 90(402, 97) *
Oct	102494, 10(44, 77)	**53357, 30(23, 14)** v	909879, 70(400, 57) *	915347, 90(402, 97) *
Nov	206210, 90(90, 45)	**75186, 10(32, 75)** v	146029, 70(64, 01) v	915347, 90(402, 97) *
Dic	97034, 90(42, 37)	**86108, 50(37, 56)** v	882598, 70(388, 55) *	915347, 90(402, 97) *

Tabla 6.5: Tamaño del modelo serializado en MBs en Weka (CV con 10 part. y 30 ejec.).

Mes	ENORA	NSGA-II	RFE	BBDD Original
Ene	$\mathbf{0,18(0,01)}$	$\mathbf{0,18(0,01)}$	$0,28(0,01)$ *	$0,34(0,01)$ *
Feb	$0,17(0,01)$	$\mathbf{0,16(0,01)}$	$0,33(0,01)$ *	$0,39(0,01)$ *
Mar	$0,21(0,01)$	$\mathbf{0,18(0,01)}$ v	$0,38(0,01)$ *	$0,41(0,01)$ *
Abr	$0,24(0,01)$	$\mathbf{0,17(0,01)}$ v	$0,29(0,01)$ *	$0,38(0,01)$ *
May	$\mathbf{0,14(0,01)}$	$\mathbf{0,14(0,01)}$	$0,18(0,01)$ *	$0,32(0,01)$ *
Jun	$\mathbf{0,14(0,01)}$	$\mathbf{0,14(0,01)}$	$0,18(0,01)$ *	$0,31(0,01)$ *
Jul	$\mathbf{0,13(0,01)}$	$\mathbf{0,13(0,01)}$	$0,18(0,01)$ *	$0,31(0,01)$ *
Ago	$0,16(0,01)$	$\mathbf{0,14(0,01)}$ v	$0,27(0,01)$ *	$0,29(0,01)$ *
Sep	$0,16(0,01)$	$\mathbf{0,13(0,01)}$ v	$0,23(0,01)$ *	$0,29(0,01)$ *
Oct	$0,12(0,01)$	$\mathbf{0,11(0,01)}$ v	$0,29(0,01)$ *	$0,29(0,01)$ *
Nov	$0,17(0,01)$	$\mathbf{0,12(0,01)}$ v	$0,15(0,01)$ v	$0,28(0,01)$ *
Dic	$\mathbf{0,11(0,01)}$	$0,12(0,01)$	$0,26(0,01)$ *	$0,27(0,01)$ *

Tabla 6.6: Tiempo en horas de CPU para el entrenamiento.

Evaluación con otros modelos de regresión

En esta sección se mide el rendimiento de las bases de datos reducidas mediante nueve modelos de regresión. Para cada BBDD reducida (meses del 1 al 12) obtenidas con ENORA, NSGA-II y RFE y para cada algoritmo de regresión (algoritmos del 1 al 9), se repiten 30 veces *cross-validation* con 10 particiones con los datos de entrenamiento utilizando el paquete Caret con el siguiente comando de control:

trainControl(method="repeatedcv", repeats=30, number=10)

Después del entrenamiento, la función "*trainControl*" elige automáticamente los mejores parámetros para cada modelo. Las Tablas 6.7, 6.8 y 6.9 muestran el RMSE para cada base de datos obtenida mediante ENORA, NSGA-II y RFE. La Tabla 6.10 muestra un resumen de los mejores resultados y del RMSE promedio obtenido con bases de datos reducidos para los métodos de regresión seleccionados (exceptuando a *Random Forest* que se muestra por separado en las tablas).

1. La evaluación con el método *Random Forest* fue mejor que con el resto de los modelos de regresión, ya que los mecanismos de selección de variables se configuraron con el método *Random Forest*.

2. ENORA obtuvo mejor promedio RMSE que NSGA-II y que RFE y para ningún método fue lo peor.

En la Figura 6.5 se muestra para cada mes el RMSE de cada método dividido entre el promedio de RMSE, proporcionando una interfaz gráfica de la visión de la desviación

Mes	M5Rules	Pcr	Relaxo	Foba	LeapForward	LeapSeq	Penalized	Ppr	Ridge
Ene	34603,87	39944,49	45966,71	37500,19	37415,92	37869,98	37379,13	**31598,47**	35275,13
Feb	17132,59	18612,82	19034,58	19286,81	18760,46	18593,09	18898,90	**15899,25**	18062,85
Mar	8159,44	8120,65	9372,00	8239,20	8118,29	7939,63	**7594,08**	8092,80	7801,88
Abr	5190,18	5361,07	5457,76	5246,77	5182,62	5046,50	5127,59	6873,05	**4879,28**
May	4507,22	5977,60	6942,10	5664,50	6119,83	5931,54	6251,55	**4076,10**	5847,55
Jun	2204,07	2498,04	2704,76	2465,42	2399,40	2570,92	2490,50	**2190,16**	2463,37
Jul	1617,91	1606,93	1592,59	**1532,73**	1570,04	1578,42	1561,18	1547,95	1537,96
Ago	1800,30	1584,01	1630,53	1596,17	1604,98	**1539,29**	1566,73	1650,14	1573,18
Sep	1823,91	1830,78	1947,19	1698,50	**1662,70**	1680,16	1705,14	1787,16	1666,77
Oct	1350,76	1332,05	1539,98	1433,86	1362,09	1317,14	1318,90	**1260,11**	1287,13
Nov	952,71	981,11	993,34	950,29	947,43	**947,08**	966,96	1021,80	963,59
Dic	989,73	1031,11	1053,37	991,17	1014,47	**956,25**	1023,27	1012,41	1032,53

Tabla 6.7: RMSE obtenido con ENORA para los métodos de regresión seleccionados.

Mes	M5Rules	Pcr	Relaxo	Foba	LeapForward	LeapSeq	Penalized	Ppr	Ridge
Ene	37661,94	41049,32	49266,68	38883,05	37587,26	37498,97	39589,37	**29116,69**	35864,30
Feb	17625,63	18643,31	19252,90	18187,14	18641,96	19016,75	18858,16	**14491,27**	18012,17
Mar	8331,65	8313,57	8189,17	**7757,02**	8042,78	8319,33	8021,13	7770,78	8263,46
Abr	5048,76	5331,02	4839,14	5107,47	5015,22	5232,16	5104,78	**4575,82**	4820,10
May	4817,66	6226,26	6724,66	6143,54	6342,04	6171,63	5980,49	**3892,11**	5775,98
Jun	2407,59	2701,44	3133,42	2394,11	2503,58	2356,52	2424,85	**2241,70**	2519,73
Jul	1657,79	1567,14	1619,78	1519,98	1557,70	1600,84	1557,53	1616,61	**1490,93**
Ago	1527,50	1602,16	1649,77	**1443,50**	1524,98	1511,47	1534,44	1477,60	1517,24
Sep	1678,39	1621,86	2125,08	1652,55	**1603,14**	1659,52	1659,12	1646,77	1773,65
Oct	1313,75	1329,80	1419,01	1333,20	1300,71	1265,15	1290,56	1260,79	**1254,84**
Nov	933,20	978,27	992,91	974,74	955,68	919,76	959,27	**916,21**	979,23
Dic	986,53	1038,50	1084,35	968,62	1023,88	1073,27	1005,11	**943,87**	1002,81

Tabla 6.8: RMSE obtenido con NSGA-II para los métodos de regresión seleccionados.

Mes	M5Rules	Pcr	Relaxo	Foba	LeapForward	LeapSeq	Penalized	Ppr	Ridge
Ene	35755,42	39855,98	41583,29	**35689,32**	36108,56	37682,69	41252,64	45814,33	40857,73
Feb	19259,82	19591,13	18606,47	18426,46	18757,90	**17718,72**	18625,31	24634,44	18630,90
Mar	9371,68	8359,13	8938,21	8242,65	8365,99	**8143,50**	8765,66	11283,30	8979,27
Abr	5545,57	**5107,87**	5142,34	5308,94	5548,14	5151,10	5534,55	6193,76	5344,75
May	5206,87	5791,20	5451,56	5229,36	**5103,85**	5642,37	5440,71	5391,23	5139,80
Jun	2290,28	2461,17	2440,11	2310,55	2283,62	2305,09	**2183,48**	2231,66	2274,90
Jul	1616,27	1537,93	1543,40	1506,39	1540,78	1530,69	1491,90	1537,00	**1479,78**
Ago	1720,56	**1597,32**	1676,43	1616,73	1610,66	1603,76	1597,38	1841,39	1618,78
Sep	1868,15	1546,99	**1546,67**	1575,85	1763,18	1599,15	1633,98	1800,92	1761,31
Oct	1391,19	1426,45	1396,30	**1333,17**	1510,83	1456,92	1494,97	1696,83	1405,26
Nov	960,20	1008,73	**905,27**	955,06	937,32	934,52	920,28	929,57	924,16
Dic	1115,10	1041,47	1042,13	987,63	971,29	**956,16**	1105,92	1360,84	1062,73

Tabla 6.9: RMSE obtenido con RFE para los métodos de regresión seleccionados.

Mes	ENORA	NSGA-II	RFE
Ene	**37505, 99**	38501, 95	39399, 99
Feb	18253, 48	**18081, 03**	19361, 24
Mar	8159, 78	**8112, 10**	8938, 82
Abr	5373, 87	**5008, 27**	5430, 78
May	5702, 00	5786, 04	**5377, 44**
Jun	2442, 96	2520, 33	**2308, 99**
Jul	1571, 75	1576, 48	**1531, 57**
Ago	1616, 15	**1532, 07**	1653, 67
Sep	1755, 81	1713, 34	**1677, 36**
Oct	1355, 78	**1307, 53**	1456, 88
Nov	969, 37	956, 58	**941, 68**
Dic	**1011, 59**	1014, 11	1071, 47
Promedio	**7143, 21**	7175, 82	7429, 16

Tabla 6.10: Resumen del mejor y del RMSE promedio.

Nombre de la variable	Total	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
Quan_2	11	•		•	•	•	•	•	•	•	•	•	•
Quan_4	9	•	•	•	•			•	•		•	•	•
Date_1	7			•				•	•	•	•	•	•
Cat_3	5			•		•	•		•			•	
Cat_209	4	•			•					•	•		•
Cat_235	4			•					•	•			
Cat_311	4		•		•				•			•	
Cat_454	4	•			•				•			•	
Cat_6	4	•			•	•	•						
Quan_8	4						•		•		•	•	

Tabla 6.11: Variables más frecuentes y meses en los que aparecen.

de cada método con respecto a la media. En la Tabla 6.11 se muestran las variables más frecuentes y los meses en los que aparecen. Y en la Tabla 6.12 se muestran los mejores RMSE por meses.

6.3.1.5. Tests estadísticos

Con el fin de encontrar posibles diferencias estadísticas significativas entre ENORA, NSGA-II, RFE y la base de datos original, testeadas con el método RF, se ha realizado el test no paramétrico de Friedman [213] con nivel de significancia $\alpha = 0,05$, con las siguientes métricas: *Ouf of bag Error*, RMSE, tamaño de la muestra y tiempo de ejecución. Anteriormente, se ha utilizado la prueba de normalidad Shaphiro-Wilk para asegurar que los datos no tienen una distribución normal.

El test de Friedman determinó que existen diferencias estadísticamente significativas

Mes	RMSE	Método	Algoritmo	Atributos seleccionados
Ene	24889,83	ENORA	Random Forest	Quan_2, Quan_4, Quan_11, Cat_6, Quan_17, Cat_126, Cat_183 Cat_198, Cat_205, Cat_209, Cat_223, Cat_238, Cat_374, Cat_381 Cat_398, Cat_450, Cat_454
Feb	11843,42	ENORA	Random Forest	Cat_1, Quan_4, Cat_4, Cat_13, Cat_164, Cat_171, Cat_240 Cat_311, Cat_334, Cat_371, Cat_375, Quan_19, Quan_21, Quan_25 Cat_494
Mar	6264,64	ENORA	Random Forest	Cat_1, Date_1, Quan_2, Quan_4, Quan_6, Cat_2, Cat_3 Cat_4, Cat_114, Cat_121, Cat_168, Cat_233, Cat_235, Cat_238 Cat_295, Cat_313, Cat_425, Cat_452, Cat_463
Abr	3812,46	ENORA	Random Forest	Quan_2, Quan_4, Quan_6, Cat_2, Cat_6, Cat_10, Cat_23 Cat_84, Quan_17, Cat_114, Cat_119, Cat_131, Cat_132, Cat_157 Cat_162, Cat_172, Cat_183, Cat_203, Cat_209, Cat_213, Cat_216 Cat_238, Cat_240, Cat_259, Cat_260, Cat_296, Cat_311, Cat_312 Cat_318, Cat_334, Cat_336, Cat_340, Cat_341, Cat_346, Cat_353 Cat_363, Cat_366, Cat_367, Cat_376, Cat_388, Cat_397, Cat_403 Cat_430, Cat_435, Cat_445, Cat_452, Cat_454, Quan_25, Cat_467
May	2635,12	ENORA	Random Forest	Quan_2, Cat_3, Cat_6, Cat_178, Cat_191, Cat_214, Cat_215 Cat_388, Cat_400, Quant_24,
Jun	1591,74	ENORA	Random Forest	Quan_2, Quan_8, Cat_3, Cat_4, Cat_6, Cat_12, Cat_205 Cat_303, Cat_491
Jul	1258,10	NSGA-II	Random Forest	Date_1, Quan_2, Quan_4, Quan_9, Quan_14, Cat_210 Cat_258, Cat_323, Cat_425, Cat_481, Cat_494
Ago	1226,30	ENORA	Random Forest	Date_1, Quan_2, Quan_3, Quan_4, Quan_8, Quan_12, Cat_3 Cat_11, Cat_15, Quan_16, Cat_178, Cat_205, Cat_235, Cat_297 Cat_311, Cat_313, Cat_316, Cat_346, Quan_21, Cat_450, Cat_451 Cat_454, Cat_483
Sep	1121,66	ENORA	Random Forest	Date_1, Quan_2, Quan_5, Quan_7, Quan_9, Cat_84, Quan_16 Quan_17, Cat_115, Cat_122, Cat_148, Cat_151, Cat_209, Cat_221 Cat_235, Cat_260, Cat_295, Cat_300, Cat_308, Cat_334, Cat_371 Cat_374, Quan_21, Cat_442, Cat_459, Cat_469, Cat_483
Oct	1004,53	ENORA	Random Forest	Date_1, Quan_2, Quan_4, Quan_8, Cat_157, Cat_191, Cat_209 Cat_213, Cat_304, Cat_376, Cat_397, Cat_403, Cat_443, Cat_481
Nov	753,16	ENORA	Random Forest	Date_1, Quan_2, Quan_3, Quan_4, Quan_8, Quan_9, Cat_2 Cat_3, Cat_12, Quan_16, Cat_114, Cat_157, Cat_168, Cat_178 Cat_183, Cat_198, Cat_203, Cat_213, Cat_214, Cat_215, Cat_226 Cat_249, Cat_300, Cat_311, Cat_340, Cat_341, Cat_397, Cat_403 Cat_436, Cat_442, Cat_454, Cat_459, Quant_24
Dic	800,53	ENORA	Random Forest	Date_1, Quan_2, Quan_4, Cat_12, Cat_223, Cat_226, Cat_235 Cat_323, Cat_371, Cat_381, Quan_21, Cat_462, Cat_498

Tabla 6.12: Mejor RMSE por cada uno de los meses.

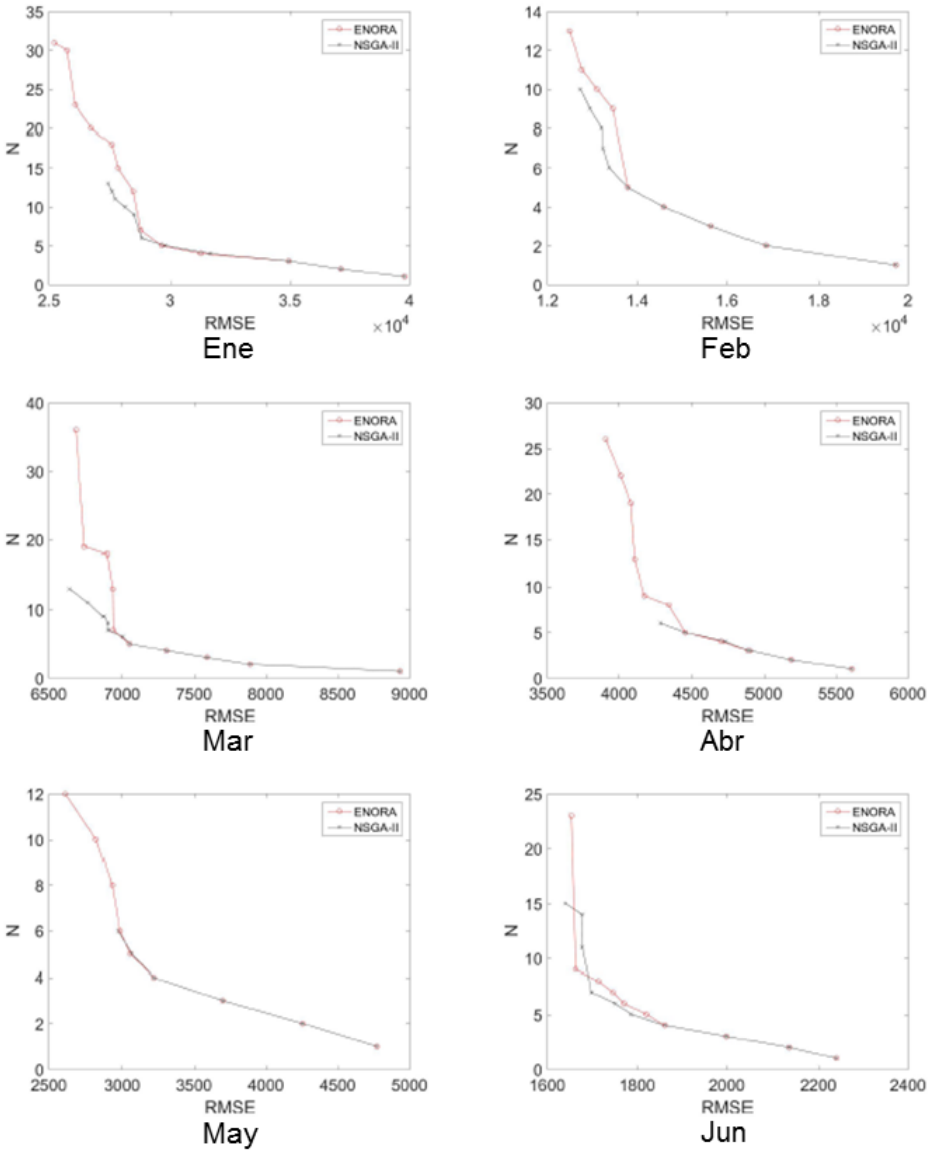

Figura 6.2: Frente de Pareto de la mejor población (Enero a Junio).

en todos los casos. Posteriormente, se realizó un test no-paramétrico Nemenyi de comparaciones múltiples post hoc para comprobar dichas diferencias. La Tabla 6.13 muestra los resultados de este análisis como son los resultados de Friedman y Nemenyi, error *Out of bag*, RMSE, tamaño de la BBDD y medidas de tiempo con el clasificador *Random Forest* (los números en negrita indican que existen diferencias significativas entre el par correspondiente de bases de datos).

Métrica	**Test de Friedmam p-value**	**Comparación de pares usando comparaciones múltiples Nemenyi con prueba post-hoc**			
Error Out Of Bag	$< 10^{-5}$		ENORA	NSGA-II	Original
		NSGA-II	0,98906	–	–
		Original	$\mathbf{5{,}6e-05}$	**0,00023**	–
		RFE-DS	0,11950	0,22910	0,11950
$\mathcal{RMSE}$	$< 10^{-5}$		ENORA	NSGA-II	Original
		NSGA-II	0,38940	–	–
		Original	$\mathbf{4{,}6e-07}$	**0,00085**	–
		RFE-DS	**0,00085**	0,11950	0,38940
Tamaño de la BBDD	$< 10^{-5}$		ENORA	NSGA-II	Original
		NSGA-II	0,68534	–	–
		Original	**0,00023**	$\mathbf{1{,}1e-06}$	–
		RFE-DS	0,16794	**0,00851**	0,16794
Tpo. entrenam.	$< 10^{-5}$		ENORA	NSGA-II	Original
		NSGA-II	0,68534	–	–
		Original	**0,00032**	$\mathbf{1{,}7e-06}$	–
		RFE-DS	0,14218	**0,00653**	0,22910
Tpo. entrenam.	$< 10^{-5}$		ENORA	NSGA-II	Original
		NSGA-II	0,68534	–	–
		Original	**0,00032**	$\mathbf{1{,}7e-06}$	–
		RFE-DS	0,14218	**0,00653**	0,22910

Tabla 6.13: Resultados de pruebas no paramétricas.

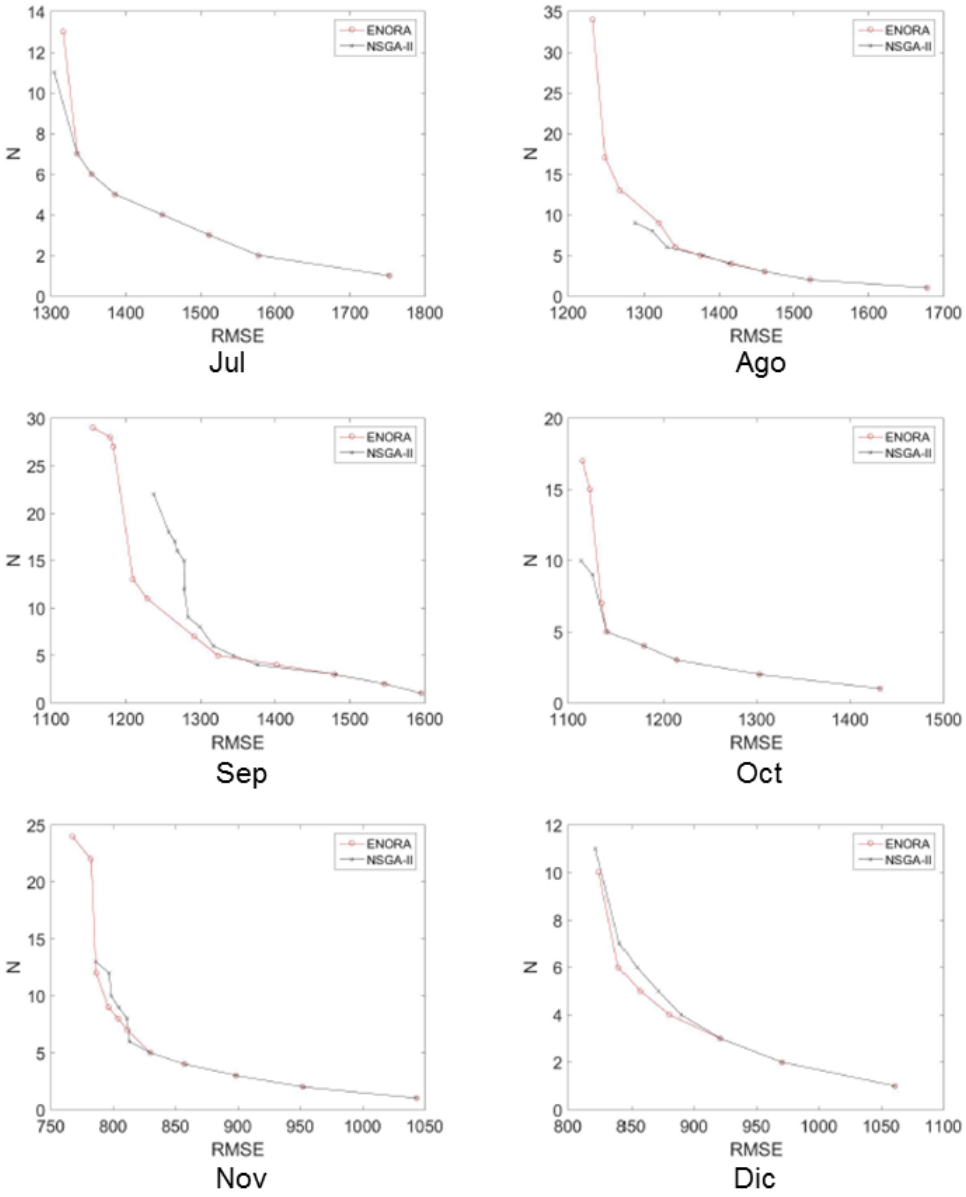

Figura 6.3: Frente de Pareto de la mejor población (Julio a Diciembre).

Nuevamente, a fin de destacar las posibles diferencias estadísticas significativas entre ENORA, NSGA-II, RFE y la base de datos original utilizando RMSE con los datos combinados para todo el año y con respecto a todos los métodos de regresión, se han realizado las pruebas de normalidad Saphiro-Wilk y el test de Friedman. Como se puede apreciar en la Tabla 6.14 (que musetra el RMSE durante todos los meses y todos los métodos de regresión seleccionados (diferentes a *random forest*), no existen diferencias estadísticas significati-

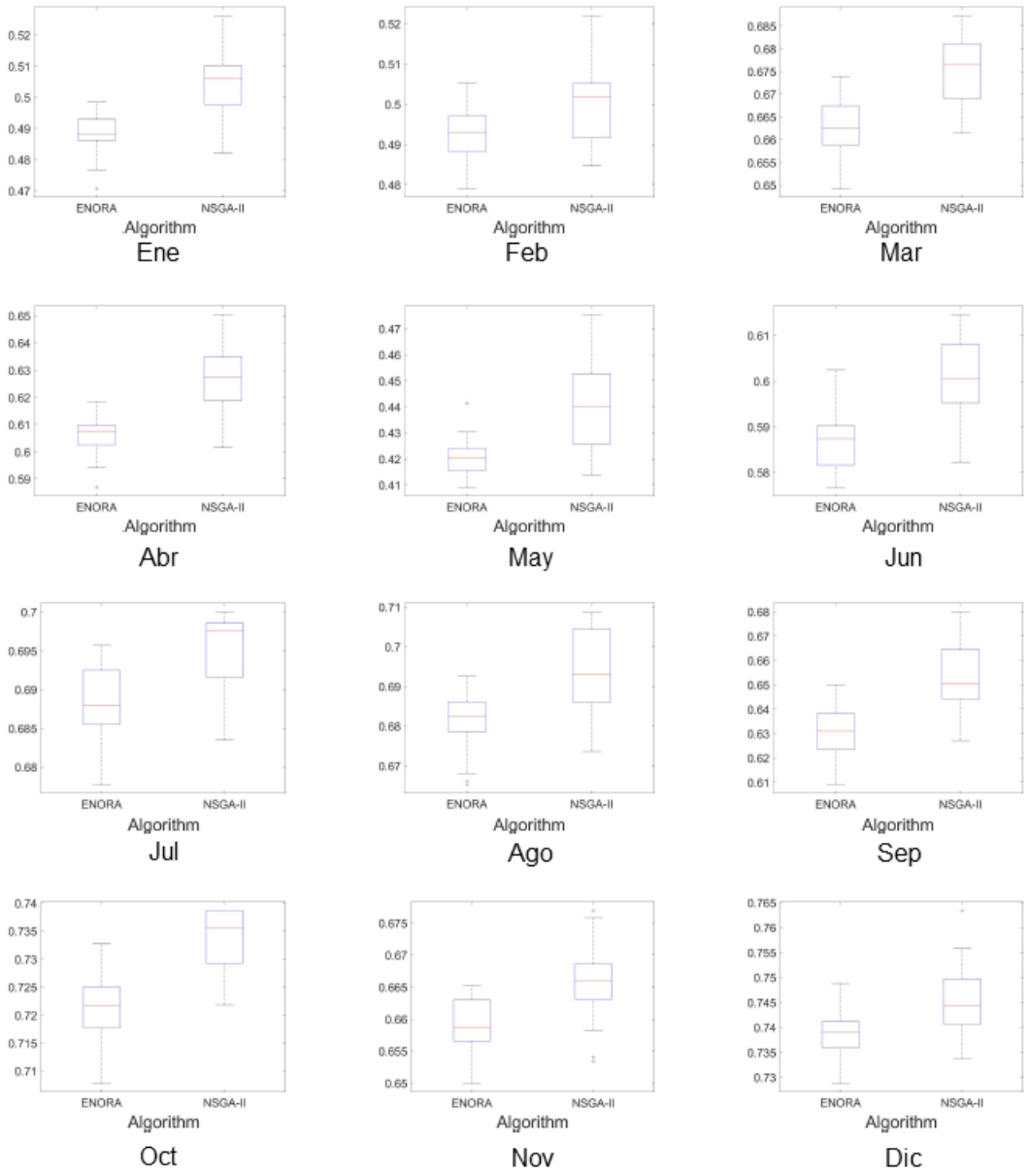

Figura 6.4: Relación de 30 hipervolúmenes entre ENORA y NSGA-II.

vas $(p - value \geq \alpha)$, por lo que no podremos realizar el test Nemenyi de comparaciones múltiples post hoc.

6.3.2. Experimento II: Selección de variables para CTR

Dentro del modelo CPC existen dos conceptos relacionados con el pago de los anunciantes que son el CPC Máximo y el CTR. El CPC Máximo representa el importe más alto que

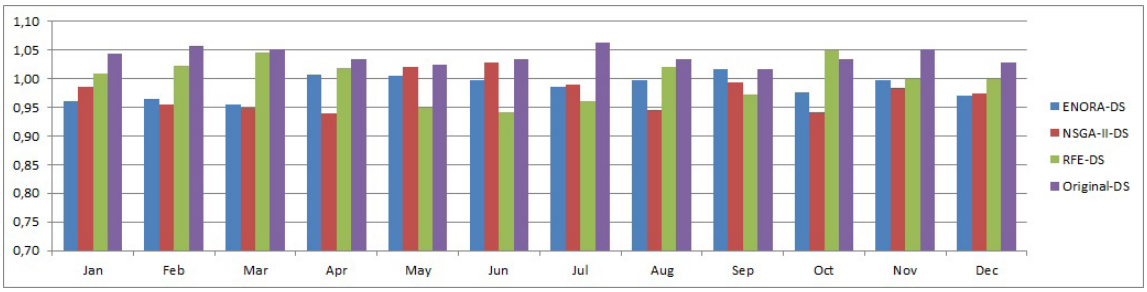

Figura 6.5: RMSE dividido entre el RMSE promedio con las BBDD.

Métrica	**Test de Friedman p-value**	**Comparación de pares mediante el test post-hoc de Nemeny de múltiples comparaciones**
$\mathcal{RMSE}$	0,5062	–

Tabla 6.14: Resultados de pruebas no paramétricas de Friedman y Nemenyi.

un anunciante está dispuesto a pagar por un clic. Sin embargo, en la mayoría de los casos se le cobra un precio menor que se denomina CPC Real. El CTR de un anuncio se define como el ratio entre el número de clics en el anuncio y el número impresiones.

Conocer con total seguridad el CTR que tendrá un determinado anuncio en el futuro no es posible, pero si se logra predecir el CTR de los anuncios con un margen de error muy pequeño se podrá elegir el anuncio que proporcione mayores beneficios [56, 186].

Existen pocas publicaciones respecto a los algoritmos utilizados por las redes publicitarias. Esto es algo lógico puesto que si se publicasen estos algoritmos las empresas perderían la ventaja competitiva. Ya que cualquiera podría copiar las investigaciones que implican tantos años de esfuerzo y tantos recursos económicos. Por otra parte, a las personas con intención de cometer fraude se les facilitaría vulnerar las redes publicitarias. Para calcular el CTR se utilizaran modelos supervisados de tipo clasificación.

6.3.2.1. Descripción de la base de datos

Para resolver el problema que se plantea se ha utilizado una base de datos proporcionada por la página web: `Kaggle.com`[1]. Las visitas se representan en una tabla donde cada fila representa la visita de un usuario y cada columna representa una característica de un usuario o de la página web. Por ejemplo, en la primera columna de la tabla se representa con “0” o con “1” si el usuario hizo clic en el anuncio. La tabla tiene trece columnas con valores

[1]Este base de datos recoge algunos parámetros de las visitas de los usuarios a la web `www.criteo.com` durante un periodo de siete días. Sobre estas visitas, se ha reducido en mayor medida el número de muestras en la que los usuarios no hicieron clic en el anuncio que los que sí lo hicieron.

de tipo entero y 26 valores de tipo *string* que representan ciertas categorías. Los valores de las categorías han sido codificados mediante un *hash* a un valor de 32 bits para garantizar la privacidad. Las filas están ordenadas cronológicamente y cuando el valor de cierto parámetro se desconoce, simplemente se deja un espacio en blanco.

Para realizar el test de prueba se utilizará un fichero con el mismo formato que la tabla de entrenamiento pero sin la columna que indica si el usuario hizo clic.

6.3.2.2. Preparación de los experimentos

Se han desarrollado numerosas herramientas para simplificar y automatizar estos procesos. Entre las herramientas de software libre más completas y con mejores resultados se encuentra R Studio. Se utilizará esta herramienta para predecir resultados con diversos métodos supervisados de clasificación. Una vez entrenados, se probará la eficacia de estos con un conjunto de entradas y posteriormente se evaluarán los modelos.

Los métodos de clasificación utilizados en los experimentos han sido: *Nearest Shrunken Centroids, Shrinkage Discriminant Analysis, CART. Partial Least Squares, Rule-Based Classifier, SVM with Radial Basis Function Kernel, AdaBoost.M1, Neural Network, Multivariate Adaptive Regression Splines, C5.0, Model Averaged Neural Network, Stochastic Gradient Boosting.*

6.3.2.3. Resultados obtenidos

Para evaluar los métodos de selección de variables utilizaremos dos métricas. La precisión o accuracy que indica el número de aciertos, y la métrica ROC. A continuación se muestran los resultados obtenidos con *ACCURACY* y con la métrica ROC en las Tablas 6.15 y 6.16 respectivamente con los distintos métodos de selección de variables.

6.3.2.4. Test estadístico paired test

El test estadístico realizado es un paired t-test (corrected) donde en cada base de datos utiliza RF (con 10 trees) y se comparan todos los métodos con la *baseline* de ENORA. Un asterisco (*) indica que la BBDD es estadísticamente peor que la de ENORA y (*v*) indica que es mejor. Si el número no está acompañado de ningún signo indica que no es ni mejor ni peor. En los tests se aplicó con *cross-validation* con 5 repeticiones y 10 particiones.

	Precisión						
Nº	**Nombre del método**	**RFE**	**PCA**	**Complete**	**GAIN**	**ENORA**	**NSGA**
1	Nearest Shrunken Centroids	0,6853	0,6856	0,6853	0,6853	0,6854	0,6853
2	Shrinkage Discriminant Analysis	0,6874	0,6905	0,6901	0,6855	0,6875	0,6853
3	CART	0,7042	0,6946	0,7042	0,7034	0,7076	0,7047
4	Partial Least Squares	0,6853	0,6902	0,6853	0,6853	0,6853	0,6846
5	Rule-Based Classifier	0,6870	0,6902	0,6828	0,7029	0,6920	0,7036
6	SVM with Radial Basis Function Kernel	0,7055	0,6957	0,7060	0,6992	0,7004	0,6923
7	AdaBoost.M1	0,7173	0,6961	0,7190	0,7143	0,7171	0,7120
8	Neural Network	0,6854	0,6977	0,6854	0,6854	0,7077	0,7061
9	Multivariate Adaptive Regression Splines	0,7095	0,6964	0,7107	0,7109	0,7115	0,7153
10	C5.0	0,7134	0,6935	0,7126	0,7042	0,7118	0,7081
11	Model Averaged Neural Network	0,6853	0,7030	0,6853	0,6853	0,7133	0,7083
12	Stochastic Gradient Boosting	0,7200	0,6989	0,7213	0,7151	0,7174	0,7145
	Promedio Acc	**0,6988**	**0,6944**	**0,6990**	**0,6981**	**0,7031**	**0,7017**

Tabla 6.15: Precisión (*Accuracy*) usando los distintos métodos de selección de variables.

	ROC						
Nº	**Nombre del método**	**RFE**	**PCA**	**Complete**	**GAIN**	**ENORA**	**NSGA**
1	Nearest Shrunken Centroids	0,6515	0,6741	0,6524	0,6332	0,6757	0,6709
2	Shrinkage Discriminant Analysis	0,6544	0,6689	0,6571	0,6109	0,6804	0,6478
3	CART	0,6344	0,6118	0,6344	0,6334	0,6351	0,6342
4	Partial Least Squares	0,6199	0,6754	0,6199	0,6185	0,6545	0,6622
5	Rule-Based Classifier	0,6468	0,6365	0,6530	0,6623	0,6617	0,6660
6	SVM with Radial Basis Function Kernel	0,6817	0,6561	0,6804	0,6559	0,6758	0,6434
7	AdaBoost.M1	0,7172	0,6764	0,7179	0,6964	0,7142	0,7008
8	Neural Network	0,5661	0,6772	0,5627	0,5641	0,6622	0,6862
9	Multivariate Adaptive Regression Splines	0,7095	0,6788	0,7100	0,6927	0,7109	0,6991
10	C5.0	0,7091	0,6516	0,7071	0,6837	0,7042	0,6807
11	Model Averaged Neural Network	0,5999	0,6820	0,6063	0,6059	0,6886	0,6914
12	Stochastic Gradient Boosting	0,7198	0,6802	0,7217	0,7014	0,7194	0,7032
	Promedio ROC	**0,6592**	**0,6641**	**0,6602**	**0,6465**	**0,6819**	**0,6738**

Tabla 6.16: ROC usando los distintos métodos de selección de variables.

	ENORA	**NSGA-II**	**RFE**	**PCA**	**GAIN**	**Completa**
Porcentaje Acierto	69,40 (1,42)	68,02 (1,47)*	69,97 (1,17)	67,92 (1,15)*	68,21 (1,11)	68,79 (1,25)
Área bajo la curva del gráfico ROC	0,67 (0,02)	0,64 (0,02)*	0,66 (0,02)	0,62 (0,02)*	0,61 (0,02)*	0,63 (0,02)*
Tamaño del Training set serializada	1123760,00	524788,00v	2250616,00*	1061639,00 v	2855919,00*	3845406,00*
Out Of Bag Error	0,3273	0,3338*	0,3277	0,3471*	0,3253	0,3262

Tabla 6.17: Test estadístico paired-test con 10 particiones y 5 repeticiones.

6.3.3. Mejora de los resultados mediante la aplicación de *Hash*

En este apartado, se aplica a las bases de datos el *hash*, que es un resumen criptográfico, para mejorar el rendimiento del modelo.

Para la base de datos de entrenamiento se utilizará el primer millón de muestras de la base de datos de Criteo y para la base de datos de pruebas se utilizará el siguiente medio millón. Para calcular el hash, se aplicará una función criptográfica que pasará el conjunto de entradas a un número entero en un rango muy grande de 0 a 100 millones. Posteriormente, se divide este valor 2^{20} que es equivalente a 1.048.576.

Al principio se tienen dos vectores de valores decimales (n, w) que se inicializan a cero y que tienen una longitud de 2^{20}. Donde *n*, es el número de veces que aparece cada índice al apliacr el hash y *w* representa los pesos para dichos índices. El valor de alfa y representa el ratio de aprendizaje de la optimización *Stochastic Gradient Descent* (SGD).

Además de actualizar el vector *n* para cada entrada de cada muestra con la fórmula $n[i] = n[i] + 1$. También se aplica el *Adaptive learning rate heuristic* para todas las entradas de todas las muestras mediante la fórmula:

$$w[i] - = (p - y) \times alfa/(\sqrt{n[i]}) + 1 \tag{6.3}$$

donde:

- $alfa$ = 0.1 (ratio de aprendizaje SGD).
- n = Vector que indica el número de veces que se ha encontrado una característica (Valor del Hash).
- w = Vector con el valor de las salidas para cada valor del Hash.
- p = Es la salida que el modelo predice para una entrada.
- y = Es la salida real de una entrada.

Una vez los pesos están actualizados la salida del modelo para una entrada se calcula mediante una función sigmoidea:

$$S(i) = 1/(1 + exp(-Wtx)) \tag{6.4}$$

donde *Wtx* representa el sumatorio de los índices del vector *w* para el conjunto de entradas de una muestra.

	Pérd. log.	Prec.	Sens.	Espec.	Recall	F1	Prec. Balan.	Nº Caract.	Caract. Selecc.
Completo	0,4667	0,7837	0,8032	0,6407	0,9425	0,7629	0,7220	39	All
Enora RMSE	**0,4811**	0,7765	0,7932	0,6300	0,9495	0,7575	0,7116	21	2,4,5,6,7,9,10,11,12,13,14,18,20,22,23,27,28,31,34,36,37
NSGA RMSE	0,5538	0,7765	0,7932	0,6300	0,9495	0,7575	0,7116	1	14
Enora Prec	**0,4904**	0,7710	0,7877	0,6113	0,9509	0,7442	0,6995	16	4,6,7,9,10,11,12,13,14,20,23,28,32,34,36,37
NSGA Prec	0,5084	0,7623	0,7741	0,5940	0,9645	0,7352	0,6840	5	7,8,9,14,16

Tabla 6.18: Valores de las tablas aplicando el *hash*.

Los resultados de aplicar esta técnica con cada una de las bases de datos se muestran en la Tabla 6.18.

6.4. Conclusiones

Se ha aplicado ENORA que es un algoritmo evolutivo multiobjetivo para la estimación de las ventas de un anuncio *online* y para calcular el CTR de un anuncio. También, se ha propuesto una metodología para integrar la selección de variables de regresión, la evaluación de los modelos y la toma de decisiones para elegir el método más satisfactorio modelo en función de un proceso *a posteriori* en cuanto al objetivo propuesto.

Se ha llegado a la conclusión de que ENORA es una buena alternativa para la selección de variables encaminadas a maximizar el RMSE y a la vez a minimizar la cardinalidad del subconjunto de los modelos. Lo que se pretende demostrar que ENORA es estadísticamente mejor con la métrica del ratio del hipervolumen y obtener modelos de regresión que son mejores que los obtenidos por NSGA-II y RFE.

Reducir el número de características de un dataset permite reducir el tiempo de construcción del modelo, reducir el tiempo de respuesta y conseguir resultados más precisos. Por estas tres razones, la selección de variables tiene una gran importancia en el aprendizaje máquina.

Esta metodología presenta una importante reducción en el número de características con respecto a los datos originales. Y además, la mejora de otras medidas importantes como el RMSE, el error *OutOfBag*, el tamaño del modelo y el tiempo de formación.

Por último, ENORA también ha tenido un buen rendimiento en la predicción del CTR. Siendo el mejor algoritmo de selección de variables de todos los utilizados. La precisión es de 0,7031 y el ROC de 0,6819, ocupando la primera posición en ambas métricas. Como trabajos futuros se podría investigar sobre los mejores métodos de selección de variables para el algoritmo *Deep Learning*. Así como aplicar los métodos de selección de variables al módulo de detección de *spam*. También se podrían aplicar nuevos tests estadísticos que nos permitan obtener conclusiones más fiables.

Otras línea de investigación más técnicas, podrían ser la integración de ENORA como estrategia de búsqueda en filtros multi-variante, y la implementación de otras herramientas

de búsqueda heurística en algoritmos (como la Optimización por enjambre de partículas) para la selección de características mejorada con la estrategia de selección de individuos de ENORA.

Parte III

Conclusiones

Capítulo 7

Conclusiones y trabajos futuros

7.1. Conclusiones

En los inicios de Internet se crearon muchas redes. Sin embargo, a pesar de la creciente demanda de la publicidad *online*, el número de redes ha ido disminuido. Las pequeñas redes se encuentran en una situación delicada debido a su carencia de sistemas eficaces para la detección de fraude y a que reciben un número bajo de visitas, lo que los hace incapaces de ofrecer campañas dirigidas a pequeños grupos.

En esta tesis doctoral se ha diseñado un modelo de intercambio de anuncios con un enfoque orientado a la satisfacción de los roles del ecosistema publicitario para que las pequeñas redes puedan competir en este mercado. Los principales objetivos de este modelo son controlar el fraude de anunciantes y editores, procurar que los anunciantes hagan campañas rentables y que se muestren anuncios, y conseguir que las redes publicitarias estén equilibradas.

Para conseguir dichos objetivos se ha utilizado un conjunto de pesos en la fórmula de selección de anuncios. Estos pesos se optimizan mediante un algoritmo genético (AG). El AG utiliza el rendimiento del modelo expresado en términos económicos como la función de aptitud, lo equivalente al *fitness*. Este valor se calcula mediante la resta entre los ingresos del modelo y el total de penalizaciones. La principal ventaja de este modelo es que incluye una metodología capaz de encontrar los valores óptimos de los pesos para cualquier configuración.

El modelo propuesto de intercambio de anuncios entre las pequeñas redes publicitarias es una solución para evitar su desaparición y también para impulsar la creación de nuevas redes publicitarias. Este modelo permitirá a las pequeñas redes ofrecer a los anunciantes campañas específicas con suficientes impresiones para que puedan competir contra las grandes redes.

Para ello, se han desarrollado una serie de pesos que intervienen en la función de selección de un anuncio y que representan la importancia que tiene cada objetivo en el desarrollo de la red. Para calcular el mejor valor para dichos pesos se ha definido una función que mide el rendimiento del modelo en términos económicos en función de un conjunto de pesos que intervienen en la selección del mejor anuncio. Esto se puede ver como una caja negra que devuelve un valor en función de un conjunto de pesos. Para encontrar la mejor combinación se ha utilizado un algoritmo genético que va creando combinaciones hasta que se cumpla el criterio de parada y el AG se quede con la mejor solución.

Antes de desarrollar el MIA, se ha comprobado que las redes publicitarias aumentan su rendimiento cuando colaboran. Para ello, se han desarrollado varios experimentos. Los experimentos muestran que cuantas más redes colaboran mejor es el rendimiento de los anuncios y la detección de fraude. También se han desarrollado varios algoritmos para intercambiar anuncios en poco tiempo y algoritmos que incluyen matrices de similitud. Así como algoritmos para que todas las redes puedan publicar anuncios de la forma más balanceada posible.

Se han utilizado varias técnicas para optimizar el intercambio de anuncios como son los hilos, los árboles con varios nodos, los árboles AVL con codificación *hash,* mediante el *software* Hadoop y utilizando matrices de similitud. También se ha utilizado tres algoritmos para repartir las visitas de una manera equilibrada: Simple, *Round Robin* y Mínimo.

Como el algoritmo de intercambio de anuncios utiliza varios módulos como son la detección de fraude y la estimación del CTR, también se han implementado estos módulos para calcular el precio de un anuncio. Aunque no se ha utilizado, también se ha desarrollado un módulo para la predicción de ventas. Este módulo se podría añadir al modelo para poder incluir el modelo de pago CPA.

Además, se ha desarrollado una metodología para calcular el valor de un anuncio en los tres modelos más extendidos como son el CPA, el CPC y el CPM. Esto tiene una ventaja muy importante y es que permite que puedan coexistir en un mismo sistema publicitario estos tres modelos de pago. Esta metodología se apoya en *Deep Learning,* que es uno de los métodos supervisados de aprendizaje máquina con mayor potencial y que ha cobrado mucho protagonismo en los últimos años.

Finalmente, se ha aplicado el algoritmo ENORA a la selección de variables tanto para la predicción de ventas como para la estimación del CTR de un anuncio. La predicción de ventas se hace mediante modelos supervisados de regresión, y la estimación del CTR se hace mediante modelos supervisados de clasificación. Se ha visto que este algoritmo da muy buenos resultados y además optimiza la cardinalidad de las características de las bases de datos.

7.2. Contribuciones principales

- Diseño de algoritmos de intercambio de anuncios y detección de fraude: Debido a que las pequeñas redes tienen pocos editores, muchas veces no se pueden satisfacer los requisitos de los anunciantes cuando hacen campañas muy segmentadas. Se ha diseñado un modelo colaborativo que mejora el rendimiento en las pequeñas redes y que aumenta su capacidad de detección de fraude. Se ha comprobado que cuanto mayor es el número de redes que colabora mayor es el porcentaje de anuncios que pueden ser cubiertos. También se han probado varios algoritmos para ejecutar el programa rápidamente. Se han utilizado los hilos, el lenguaje Pig Latin, y distintos tipos de árboles. Los árboles han sido los que mejor rendimiento han dado.

  Para mejorar la detección de fraude se ha diseñado un entorno colaborativo en el que cada una de las redes avisa al resto cuando detecta una IP de un determinado *click-bot* o un usuario malintencionado comprobando cómo aumenta la detección de fraude cuando las redes colaboran entre sí. Para detectar fraude se han utilizado las técnicas de *captcha* y la de anuncios irrelevantes.

- Algoritmos de reparto de visitas: También se han diseñado tres algoritmos para el reparto de anuncios (el simple, el *Round Robin* y el de visitas mínimas) para balancear el número de anuncios publicados en cada una de las redes que colaboran. Es decir, para que el número de anuncios se reparta equitativamente entre las redes, de forma que todas redes se beneficien al colaborar. Se ha comprobado que el de visitas mínimas es el mejor de los tres, aunque precisa de una lista con el historial de anuncios publicados.

- Diseño de un modelo de intercambio de anuncios con un nuevo enfoque: En este modelo, la función de selección de anuncios evalúa varios objetivos que se consideran necesarios para un funcionamiento adecuado del ecosistema publicitario. Los objetivos que se han tenido en cuenta son los ingresos del modelo, la satisfacción de anunciantes, editores y redes de publicidad, y el fraude tanto de anunciantes como de editores.

  Para garantizar que todos los objetivos se cumplen se han definido un conjunto de penalizaciones. Para ello, en la fórmula de selección de un anuncio se utiliza una variable para representar cada objetivo y un peso para modelar la importancia de cada objetivo. La suma de los pesos ha de ser "1" y los pesos se han optimizado mediante un AG de acuerdo con el rendimiento del modelo. El rendimiento del modelo es igual a los ingresos menos la suma de un conjunto de penalizaciones que se aplican cuando un

objetivo no se cumple. También se explican todos los pasos necesarios para desarrollar el MIA. Se enumeran los principales objetivos para el correcto funcionamiento del MIA y las medidas adoptadas para defenderse contra el fraude *online*. También se explican las penalizaciones para garantizar que se alcancen los objetivos y las métricas utilizadas para medir el rendimiento del MIA.

- Diseño de una metodología para calcular el valor de un anuncio: El sistema que se ha diseñado tiene dos entradas y una salida. Las entradas son la visita del usuario y los datos sobre el anunciante. La salida es la estimación del valor del anuncio expresado en dólares. Mediante los métodos de aprendizaje supervisado *Deep Learning* ha modelado el comportamiento de eventos relacionados con la publicidad *online* con mayor precisión. Los métodos *Deep Learning* tienen un sistema de capas intermedias que permite a estos modelos tener un mayor nivel de abstracción que el resto de métodos supervisados tradicionales.

- Aplicación del algoritmo de selección de variables ENORA: Se ha aplicado un algoritmo llamado ENORA. ENORA es una estrategia de búsqueda evolutiva multiobjetivo para identificar soluciones de pareto óptimas simultáneamente, teniendo en cuenta los criterios de exactitud y la cardinalidad del subconjunto. Esta estrategia de búsqueda puede ser utilizada en métodos de selección de tipo filtro y de tipo *wrapper*, pero se ha utilizado como método de aprendizaje un método *wrapper* con *Random Forest*. ENORA ha dado los mejores modelos tanto para predicción de ventas (regresión) como para la predicción del CTR (clasificación).

7.3. Publicaciones científicas derivadas de las tesis

Las publicaciones se han agrupado según sean revistas internacionales o artículos de congreso.

7.3.1. Revistas internacionales

1. **International Journal of Engineering and Management Research (IJEMR):** "Miralles-Pechuán, Luis., Ballester, E. M. y García, José M. Online Advertising and the CPA Model: Challenges and Opportunities (2014). *International Journal of Engineering and Management Research,* http://www.ijemr.net, 4, 324-334".

   Este artículo está relacionado con el capítulo 2 en el que se hace una introducción al estado del arte. Este artículo fue redactado al inicio de la tesis durante el estudio de

los principales modelos de pago en la publicidad online. El objetivo del artículo es comparar los modelos de pago entre sí y destacar las ventajas del modelo CPA.

La publicidad en Internet se ha convertido en uno de los canales esenciales para cualquier campaña publicitaria. Como era de esperar, el volumen de negocio de la publicidad en Internet se ha convertido en una atracción para muchos ciberdelincuentes que tratan de enriquecerse a través de la estafa. Además, los crímenes cometidos en Internet son más difíciles de sancionar debido a la complejidad en la detección de crímenes, a que se cometen desde otro país y a la volatilidad de las pruebas. Las principales amenazas son los famosos *click-bots*, que son un software que simula el comportamiento humano de navegación y que genera clics ilegales. Para evitar el fraude por clic se desarrolló el sistema CPA. En este sistema los usuarios pagan por acciones en lugar de pagar por clics. A lo largo de este artículo se discuten las principales ventajas y desventajas del modelo CPA, los principales tipos de fraudes que pueden ocurrir y cómo detectarlos.

2. **Soft Computing**: *"*Miralles-Pechuán, Luis., Rosso, Dafne., Jiménez, Fernando., García, José. M. (2016). A methodology based on Deep Learning for advert value calculation in CPM, CPC and CPA networks*. Soft Computing. Springer"*

   Este artículo aborda el capítulo 5 de la tesis. Puede ser considerado la tercera parte de la tesis, pues en este punto se desarrollan las nuevas ideas. En esta parte se ha diseñado un mecanismo para calcular el valor de un anuncio en los tres modelos de pago de la publicidad *online*.

   En esta investigación se propone una metodología para el cálculo del valor de un anuncio en las redes CPM, CPC y CPA. Estimar con precisión este valor aumenta los ingresos de las redes publicitarias mediante la selección del anuncio más rentable. Al aumentar los ingresos, los editores pueden estar mejor pagados y se pueden ofrecer mejores servicios a los anunciantes. Para el desarrollo de esta metodología se propone un sistema basado en métodos tradicionales de aprendizaje máquina y *Deep Learning*. Los experimentos realizados permiten concluir que el DL es un método supervisado muy eficiente para la clasificación de anuncios *spam* y para la estimación del CTR. DL ha demostrado un rendimiento aceptable en la predicción de ventas.

3. **Neuro Computing:** "Jiménez, F., Sánchez, G., García, José M., Sciavicco, G., Miralles-Pechuán, Luis. (2016). Multi-objective Evolutionary Feature Selection for Online Sales Forecasting. *Neurocomputing. Elsevier*"

   Este artículo abarca el capítulo 6 y se realizó a continuación del anterior. En este artículo se aplica la selección de variables para lograr mejores resultados en la predic-

ción de ventas. Este artículo se centra únicamente en mejorar la predicción de ventas mediante el algoritmo ENORA. Este algoritmo se compara con otros dos algoritmos conocidos de tipo *wrapper* que son el RFE y el NSGA-II.

En la predicción de ventas se utiliza información como: las ventas pasadas, las características de los productos o la situación del mercado. Predecir las ventas a corto o a largo plazo, puede ser utilizado para la estrategia de la empresa en el futuro y puede suponer algo vital para la supervivencia de una empresa. Si se predice que se va a vender más de lo real, la empresa puede endeudarse y si se predice menos, se puede desperdiciar una oportunidad de negocio. Para predecir ventas se emplean los métodos supervisados de ML y también se aplican los métodos de selección de variables. En este artículo se compara el rendimiento de ENORA, un novedoso método de selección de variables con los famosos métodos RFE y NSGA-II.

4. **Neural Computing and Applications (NCAA):** "Miralles-Pechuan, Luis., Rosso, Dafne., Jimenez, Fernando., Sánchez, Gracía. (2017). A Novel Advertising Exchange Model for Small Online Advertising Networks. *Neural Computing and Applications. Springer*" **(Artículo en revisión).**

   Este artículo se aborda en el capítulo 4 de la tesis. A lo largo de este artículo se propone un modelo de intercambio de anuncios bajo una nueva perspectiva como se ha comentado ampliamente en el capítulo. Este artículo fue redactado seguidamente al artículo anterior aunque en la tesis no estén en orden cronológico. En este artículo se aprovecha el conocimiento del estado del arte para crear un modelo de intercambio de anuncios basado en un nuevo enfoque.

   Las redes de publicidad pequeñas han ido desapareciendo poco a poco pues han sido incapaces de ofrecer a los anunciantes campañas específicas. En esta investigación se desarrolla un modelo de intercambio de publicidad entre las redes de publicidad pequeñas para que puedan competir contra las grandes redes. Para ello, se definen los objetivos necesarios para un intercambio de publicidad óptima y un conjunto de normas destinadas a luchar contra el fraude *online*. Después, se define una función para clasificar anuncios mediante la evaluación de todos los objetivos a través de un conjunto de pesos. Por último, se desarrolla una metodología para calcular los pesos óptimos para la función de selección de un anuncio a través de un algoritmo genético. Esta metodología optimiza el peso de la función según el rendimiento del modelo expresado en términos económicos.

7.3.2. Actas de congresos:

1. **Mexican International Conference on Artificial Intelligence (MICAI 2014):** "Miralles-Pechuán, Luis. y García, José M. Bringing Networks Together to Improve Advertising Performance. *Research in Computing Science, 2014, 86, 63-75"*.

   Este artículo se redactó en los comicios de la tesis doctoral y está relacionado con el capítulo 3. El objetivo del artículo es comprobar que las pequeñas redes trabajando juntas pueden crear un modelo más competitivo para enfrentarse a las redes grandes.

   Creemos que las pequeñas redes trabajando juntas pueden crear una solución más competitiva para poder competir con las redes más grandes, tanto en el rendimiento de los anuncios como en la detección de fraude. Además, hemos diseñado algoritmos para distribuir uniformemente las visitas en varias redes, y además hemos utilizado la desviación media como parámetro para comparar resultados.

2. **MICAI 2014:** "Miralles-Pechuán, Luis., Gómez, Claudia. y Martínez-Villaseñor, Lourdes. Ad Exchange Optimization Algorithms on Advertising Networks. *Research in Computing Science, 2014, 84, 77-88"*.

   Este artículo se redactó a continuación del artículo anterior y también está relacionado con el capítulo 3. El objetivo es probar varias estrategias para elegir el mejor anuncio de entre todos los anunciantes en función del costo computacional y del número de comparaciones.

   La publicidad *online* ha experimentado un gran crecimiento en los últimos años. Los anunciantes han conseguido mejores resultados con campañas dirigidas a audiencias más específicas. Las redes de anuncios con pocas visitas no son capaces de crear dichas campañas y por lo tanto, deberían converger hacia modelos más colaborativos. Este modelo se puede concebir como un mercado en el que millones de anunciantes compiten por mostrar un anuncio. En la selección del mejor candidato, los algoritmos deben ser capaces de procesar los requisitos de todos los anunciantes en escasas décimas de segundo.

   Para afrontar este problema se han desarrollado algoritmos utilizando técnicas como hilos, árboles AVL con *hash*, árboles con varios nodos o Hadoop. A lo largo de este artículo se muestran los resultados obtenidos a partir de cada algoritmo, un análisis comparativo del rendimiento y algunas conclusiones. También se han propuesto algunas líneas futuras de trabajo.

3. **Congreso Mexicano de Inteligencia Artificial (COMIA 2014):** "Miralles-Pechuán, Luis. y Ponce, Hiram. Predicción del CTR de los anuncios de Internet usando redes

orgánicas artificiales. *Research in Computing Science, 2015, 93, 23-32"*.

Este artículo está relacionado con los capítulos 5 y 6, y fue realizado después de introducirnos en el *Machine Learning* y el manejo de R Studio, lo que se considera la segunda etapa de la tesis.

En este artículo se propone mejorar la predicción del CTR. Para que las redes publicitarias aumenten sus ingresos es necesario darle prioridad a los anuncios más rentables. El factor más importante en la rentabilidad de un anuncio es el CTR y para predecir el CTR, se han entrenado varios modelos de clasificación supervisados y se ha comparado su rendimiento con las Redes Orgánicas Artificiales (ROA). La conclusión es que estas redes son una buena solución para predecir el CTR de un anuncio.

4. **MICAI 2015:** "Ponce, Hiram. Miralles-Pechuán, Luis y Martínez-Villaseñor, Lourdes. Artificial Hydrocarbon Networks for Online Sales Prediction. *Mexican International Conference on Artificial Intelligence, 2015, 498-508"*.

   Esta investigación guarda relación con los capítulos 5 y 6, y se realizó en la segunda etapa de la tesis. En esta etapa se empieza a utilizar ML para problemas de regresión.

   Este artículo trata de mejorar la predicción de ventas mediante el uso de redes orgánicas artificiales. Las ventas a través de internet han crecido mucho en la última década. Con el fin de hacer frente a esta alta demanda, la predicción y la publicidad en línea se han vuelto muy importantes.

   Para hacer una inversión adecuada es necesario tener un modelo que relacione determinadas características del producto con el número de ventas que se producirán en el futuro. En este trabajo se presenta un análisis comparativo de varias técnicas de *Machine Learning* frente a un nuevo modelo de aprendizaje supervisado llamado Redes Orgánicas Artificiales. Este método es un nuevo tipo de aprendizaje automático que ha demostrado adaptarse muy bien a un amplio espectro de problemas de regresión y de clasificación.

7.4. Publicaciones científicas relacionadas con las tesis

Las publicaciones se han agrupado según sean revistas internacionales y artículos de congreso.

7.4.1. Revistas internacionales

1. **Sensors:** *"*Ponce, Hiram., Martínez-Villaseñor, Lourdes. y Miralles-Pechuán, Luis. (2016). A Novel Wearable Sensor-Based Human Activity Recognition Approach Using Artificial Hydrocarbon Networks. *Sensors, 16 (7), 1033"*.

   El reconocimiento de actividades humanas ha cobrado un gran interés en muchas áreas debido a que ayuda a proporcionar servicios proactivos y personalizados. No obstante, el reconocimiento de actividades no es una tarea fácil. Los diferentes tipos de ruido en los sensores dañan los datos con frecuencia y obstaculizan el proceso de clasificación en el reconocimiento de actividades humanas. En este trabajo se presentan las redes artificiales de hidrocarburos con un enfoque novedoso y adecuado para las actividades físicas, así como un método robusto y con buena tolerancia al ruido.

2. **Sensors:** *"*Ponce, Hiram., Miralles-Pechuán, Luis. y Martínez-Villaseñor, Lourdes (2016). A Flexible Approach For Human Activity Recognition Using Artificial Hydrocarbon Networks. *Sensors, 16, Advances in Artificial Intelligence: Selected Papers from MICAI 2013, 2014 and 2015—12th, 13th and 14th Mexican International Conferences on Artificial Intelligence"*.

   En este trabajo se aborda uno de los principales retos del reconocimiento de actividades humanas: la flexibilidad. El objetivo es presentar las redes artificiales de hidrocarburos con un enfoque novedoso y flexible en el reconocimiento de actividades humanas. Estos resultados demuestran que dichas redes son clasificadores suficientemente flexibles como para ser utilizados al detectar nuevas actividades humanas dependientes e independientes de los usuarios.

3. **Aranzadi:** *"*Otero, J. M. M., y Miralles-Pechuán, Luis (2014). Fraudes en la publicidad en Internet: tipología y tratamiento jurídico. *Revista Aranzadi de derecho y nuevas tecnologías, (34), 67-90"*.

   La publicidad en Internet constituye la principal fuente de ingresos de los editores de páginas web. Los principales modelos de pago a los editores son tres: *Cost-per-mille*, *Cost-per-click* y *Cost-per-action*. En estos modelos se han desarrollado varias técnicas fraudulentas que pueden ser realizadas por personas o por robots. Estas técnicas intentan falsear datos para ganar dinero a costa de los anunciantes. El artículo analiza los fraudes más extendidos y ofrece la respuesta jurídica que se puede presentar a los mismos.

7.4.2. Actas de congresos

1. **Congreso Mexicano de Inteligencia Artificial (COMIA) 2014:** *"*Miralles-Pechuán, Luis., Pelayo, D. R. y Brieva, J. Reconocimiento de dígitos escritos a mano mediante métodos de tratamiento de imagen y modelos de clasificación. *Research in Computing Science, 2015, 93, 83-94"*.

   En este artículo se habla del Reconocimiento Óptico de Caracteres (ROC) que es una línea de investigación dentro del procesamiento de imágenes para la que se han desarrollado muchas técnicas y metodologías. Su objetivo principal consiste en identificar un carácter, ya sea letra o número, a partir de una imagen digitalizada que se representa como un conjunto de píxeles. En este trabajo se realiza para el ROC un proceso iterativo que consta de cinco fases: redimensionar la imagen, tratamiento de la imagen, selección de variables, construcción y evaluación del modelo, y muestra de resultados. Para ello se aplican diversos métodos supervisados de aprendizaje automatizado. Entre los modelos de clasificación destaca *Deep Learning* por su novedad y por su enorme potencial.

2. **Congreso Ubiquitous Computing and Ambient Intelligence (UCAmI) 2015:** *"*Ponce, H., Martínez-Villaseñor, Lourdes y Miralles-Pechuán, Luis. Comparative Analysis of Artificial Hydrocarbon Networks and Data-Driven Approaches for Human Activity Recognition. *International Conference on Ubiquitous Computing and Ambient Intelligence, 2015, 150-161"*.

   En los últimos años, la informática y la tecnología de los sensores han supuesto un gran avance que contribuye al desarrollo efectivo de la actividad humana en el reconocimiento de actividades. En muchas aplicaciones, la clasificación de posturas corporales y movimientos ayuda al desarrollo de sistemas de salud que mejoran la calidad de vida de los discapacitados y de los ancianos. En este trabajo se presenta un análisis comparativo de datos en el reconocimiento de actividades mediante las redes orgánicas artificiales. En este artículo se ha comprobado que estas redes son adecuadas para una eficiente clasificación de las posturas corporales y los movimientos. Para ello, se ha comparado el desempeño de estas redes con el de otros métodos tradicionales de *machine learning*.

7.5. Futuras líneas de investigación

A continuación se describen sucintamente varias líneas de investigación que se han abierto mientras se realizaba la tesis, y que merecería la pena desarrollarlos en el futuro.

- Capítulo 3: Una mejora en los algoritmos de intercambio de anuncios sería quedarnos con el algoritmo ordenado por frecuencia y en lugar de ordenar cada nueva campaña, aplicar el algoritmo de ordenación cada 1.000 campañas. Con esta fórmula se podría crear un árbol y comparar cada visita con el árbol formado de los anunciantes.

- Capítulo 4: En el modelo de intercambio de anuncios se podría encontrar el valor real de las penalizaciones. Este valor podría encontrarse construyendo modelos complejos de simulación o haciendo pruebas en escenarios reales. Como futuras líneas de investigación, se podría también incluir en el MIA los modelos de pago CPM y CPA.

  Además, se podrían desarrollar nuevos módulos que permitan a las redes cooperar entre sí con el objetivo de mejorar la detección de fraude. Para ello, se podría intercambiar información como el CTR de las páginas, el CTR de los anuncios o los patrones de comportamiento de los usuarios. Esto podría hacerse recolectando muestras del comportamiento para su posterior análisis mediante modelos de aprendizaje automático. Cuantas más muestras y cuanta mayor sea su calidad, más precisos serán los modelos construidos.

  Otras investigaciones implicarían desarrollar un modelo escalable, es decir, en lugar de construir un modelo con 10 redes de 10 anunciantes y de 100 editores, se podría desarrollar un modelo con 1.000 redes de 10.000 anunciantes y 100.000 editores. Para ello, se podría considerar replicar el MIA utilizando una arquitectura distribuida en lugar de una arquitectura centralizada. Estos módulos podrían sincronizarse con el intercambio de información dentro de las redes, por lo que las variables se actualizan y por lo que se optimiza el tiempo de respuesta para cada usuario.

  Para ello, se requiere un protocolo de comunicación entre los diferentes sistemas de intercambio de anuncios. Este protocolo transferirá la información necesaria dentro del MIA para optimizar los beneficios económicos de cada red, para evitar el fraude y, por último, para mantener el nivel de satisfacción de todas las partes involucradas en el modelo. Los siguientes pasos a la creación del modelo: encontrar un protocolo de comunicación entre los MIA y encontrar el hardware para poder desarrollar el MIA.

- Capítulo 5: Respecto al cálculo del valor de un anuncio, una interesante línea de mejora para obtener mayor rendimiento sería hacer varias réplicas del modelo para equilibrar la carga de trabajo y, por tanto, reducir el tiempo de respuesta. El sistema desarrollado podría mejorarse añadiendo mecanismos con el fin de detectar el fraude de un anuncio *online*. Esto se podría hacer añadiendo tres nuevos módulos que se podrían llamar A, B y C.

El módulo A se utilizaría en las redes CPM para medir la probabilidad de que el anuncio sea falso. El módulo B se puede emplear en la redes CPC para evaluar la probabilidad de que un clic sea falso. El módulo C se puede utilizar en las redes CPA para calcular la probabilidad de que una acción sea fraudulenta. Un ejemplo de acción fraudulenta podría ser que un editor le diga un amigo que rellene un formulario con la única intención de obtener una comisión.

- Capítulo 6: Se ha considerado utilizar la estrategia de búsqueda integrada en el método de selección de variables ENORA de tipo filtro con el fin de evitar la dependencia del método de selección en el modelo de regresión. Otra línea de investigación consistiría en diseñar una nueva estrategia de búsqueda para la selección de variables basada en la optimización multiobjetivo con partículas *Swarm*. También se podría aplicar el método de selección de variables al nuevo algoritmo *Deep Learning* para ver si se incrementa su rendimiento. Así como optimizar el módulo de detección de anuncios de tipo *spam*.

Lista de siglas

AG Algoritmo Genético

ANN Artificial Neural Network - Red neuronal artificial

ARIMA Autoregressive Integrated Moving Average - Modelo autorregresivo integrado de media móvil

AVL Adelson – Velskii – Landis

BBDD Base de datos

BCI Brain Computer Interface - Interfaz cerebro computadora

CART Classification And Regression Trees - Árboles de clasificación y de regresión.

CART Classification and regression tree - Árboles de clasificación y regresión

CD Compact Disc - Disco compacto

CE Computación Evolutiva

CPA Cost per action - Costo por acción

CPC Cost per click - Costo por cada clic

CPCPV Cost Per Click Play View - Costo por clic para visualizar

CPCV Cost Per Completed View - Costo por visualización completa

CPM Cost per mille - Costo por cada mil

CPU Central Processing Unit - Unidad central de procesamiento

CPV Cost per visitor - Costo por visitante

CTR Click through rate - Ratio de clics

CV Cross validation - Validación cruzada

DBN Deep Belief Network - Red de creencia profunda

DL Deep Learning - Aprendizaje profundo

DT Decision tree - Árbol de decisión

ENORA Evolutionary Non dominated Radial slots based Algorithm - Algoritmo evolutivo no dominado basado en espacios radiales

GIF Graphics Interchange Format - Formato de gráficos para intercambiar

GSP Global second price - Segundo precio global

HTTP Hypertext Transfer Protocol - Protocolo de transferencia de hipertexto

IAB Interactive Advertising Bureau - Oficina de publicidad interactiva

ID3 Iterative Dichotomiser 3 - Dicotomizador interactivo 3

IG Informacion Gain - Ganancia de información

IP Internet Protocol - Protocolo de Internet

JPG Joint Photographic Experts Group - Grupo de expertos en fotografia

KDD Knowledge Discovery in Databases - Conocimiento en Bases de datos

KMCP Kernel-based multiple criteria programming - Programación multi criterio basada en núcleo múltiple

KMCR Kernel Based Multiple Criteria Regression - Criterio multiple basado en núcleo para regresión

MBN-EDA Multi-dimensional Bayesian network (MBN) Estimation of distribution algorithms (EDAs) - Red bayesiana multidimensional Estimación de los algoritmos de distribución

MCLP Multi Criteria Linear Programming - Programación lineal multicriterio

MCLR Multi Criteria Linear Regression - Regresión lineal multicriterio

MIA Modelo de intercambio de anuncios

ML Machine Learning - Aprendizaje Máquina

MO Multiobjective optimization - Optimización multiobjetivo

MOEA Multi-Objective Evolutionary Algorithm - Algoritmo evolutivo multi objetivo

NN Neural Networks - Redes neuronales

NSGA-II Nondominated sorting genetic algorithm - Algoritmo Genético de Clasificación No dominado

OOB Out Of Bag - Fuera de la bolsa

PCA Principal Component Analysis - Análisis de componentes principales

PPV Pay per view - Pago por impresión

RF Random Forest - Selvas Aleatorias

RFE Recursive Feature Elimination - Eliminación recursiva de variables

RMSE Root Mean Squared Error - Promedio de la raíz cuadrada del error cuadrático

RTB Real Time Bidding - Apuestas en tiempo real

SEM Search Engine Marketing - Marketing en buscadores web

SEM Search engine marketing - Promoción en motores de búsqueda

SEO Search engine optimization - Optimización para los motores de búsqueda

SERP Search Engine Results Pages - Páginas de resultados de los motores de búsqueda

SIA Sistema de intercambio de anuncios

SMS-EMOA Simetric selection Evolutionary multiobjective optimization algorithms - Selección simétrica Algoritmos evolutivos de optimización multiobjetivo

SMTP Simple Mail Transfer Protocol - Protocolo para transferencia simple de correo

SO Sistema Operativo

SVM Support Vector Machine - Máquina de vectores de soporte

SVR Support vector regression - Regresión de máquina de vectores

URL Uniform Resource Locator - Localizador de recursos uniforme

Referencias

[1] S Muthukrishnan. Ad exchanges: Research issues. In *Internet and network economics*, pages 1–12. Springer, 2009.

[2] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. Real-time bidding for online advertising: measurement and analysis. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, page 3. ACM, 2013.

[3] Shalinda Adikari and Kaushik Dutta. Real time bidding in online digital advertisement. In *New Horizons in Design Science: Broadening the Research Agenda*, pages 19–38. Springer, 2015.

[4] Claude Castelluccia, Lukasz Olejnik, and Tran Minh-Dung. Selling off privacy at auction. In *Network and Distributed System Security Symposium (NDSS)*, 2014.

[5] Ruggiero Cavallo, R Preston Mcafee, and Sergei Vassilvitskii. Display advertising auctions with arbitrage. *ACM Transactions on Economics and Computation*, 3(3):15, 2015.

[6] Santiago R Balseiro, Jon Feldman, Vahab Mirrokni, and S Muthukrishnan. Yield optimization of display advertising with ad exchange. *Management Science*, 60(12):2886–2907, 2014.

[7] Fiona Ellis-Chadwick and Neil F Doherty. Web advertising: The role of e-mail marketing. *Journal of Business Research*, 65(6):843–848, 2012.

[8] John A Bargh and Katelyn YA McKenna. The internet and social life. *Annu. Rev. Psychol.*, 55:573–590, 2004.

[9] Sergei N Dorogovtsev and José FF Mendes. *Evolution of networks: From biological nets to the Internet and WWW*. Oxford University Press, 2013.

[10] Barry M Leiner, Vinton G Cerf, David D Clark, Robert E Kahn, Leonard Kleinrock, Daniel C Lynch, Jon Postel, Larry G Roberts, and Stephen Wolff. A brief history of the internet. *ACM SIGCOMM Computer Communication Review*, 39(5):22–31, 2009.

[11] Price Water House Coopers. Iab internet advertising revenue report. *URL http://www. iab. net/insights_research/industry_data_ and_landscape/adrevenuereport*, 2014.

[12] Qing Cui, Feng-Shan Bai, Bin Gao, and Tie-Yan Liu. Global optimization for advertisement selection in sponsored search. *Journal of Computer Science and Technology*, 30(2):295–310, 2015.

[13] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. Technical report, National Bureau of Economic Research, 2005.

[14] M Palanivel and R Uthayakumar. Finite horizon eoq model for non-instantaneous deteriorating items with price and advertisement dependent demand and partial backlogging under inflation. *International Journal of Systems Science*, 46(10):1762–1773, 2015.

[15] Lampros C Stavrogiannis, Enrico H Gerding, and Maria Polukarov. Auction mechanisms for demand-side intermediaries in online advertising exchanges. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1037–1044. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[16] Sebastian Angel and Michael Walfish. Verifiable auctions for online ad exchanges. In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 195–206. ACM, 2013.

[17] Auguste Kerckhoffs. *La cryptographie militaire*. University Microfilms, 1978.

[18] Daniel C Fain and Jan O Pedersen. Sponsored search: A brief history. *Bulletin of the American Society for Information Science and Technology*, 32(2):12–13, 2006.

[19] Jonathan Levin and Paul Milgrom. Online advertising: Heterogeneity and conflation in market design. *The American Economic Review*, pages 603–607, 2010.

[20] Alexander Tuzhilin. The lane's gifts v. google report. *Official Google Blog: Findings on invalid clicks, posted*, pages 1–47, 2006.

[21] Markus Jakobsson and Zulfikar Ramzan. *Crimeware: understanding new attacks and defenses*. Addison-Wesley Professional, 2008.

[22] David J Reibstein. What attracts customers to online stores, and what keeps them coming back? *Journal of the academy of Marketing Science*, 30(4):465–473, 2002.

[23] Miniwatts Marketing Group. Internet world stats. `http://www.internetworldstats.com/stats.htm`, 2015.

[24] David Lange and Steffen Manes. *War for Growth: How to make it in the Chinese Internet Market: Lessons learned from Groupon, Google, Yahoo and others*. Wagner Verlag sucht Autoren, 2013.

[25] StatCounter. Statcounter. `http://gs.statcounter.com/#browser-ww-monthly-201507-201512-bar`, 1999-2016.

[26] IAB internet advertising revenue report. Iab internet advertising revenue report - iab france. `http://www.iab.com/wp-content/uploads/2015/10/IAB_Internet_Advertising_Revenue_Report_HY_2015.pdf`, 1999-2016.

[27] Avi Goldfarb and Catherine Tucker. Online display advertising: Targeting and obtrusiveness. *Marketing Science*, 30(3):389–404, 2011.

[28] WW Moe. Targeting display advertising. *London, UK: Advanced Database Marketing: Innovative Methodologies & Applications for Managing Customer Relationships*, 2013.

[29] Eugene Sivadas, Rajdeep Grewal, and James Kellaris. The internet as a micro marketing tool: targeting consumers through preferences revealed in music newsgroup usage. *Journal of Business Research*, 41(3):179–186, 1998.

[30] Nikolay Archak, Vahab Mirrokni, and S Muthukrishnan. Budget optimization for online advertising campaigns with carryover effects. In *Sixth Ad Auctions Workshop*. Citeseer, 2010.

[31] Carl Dunham, Alan Trzcinko, Brian McCarthy, and James Beriker. Cost-per-action search engine system, method and apparatus, July 31 2002. US Patent App. 10/210,677.

[32] Matthew Richardson, Ewa Dominowska, and Robert Ragno. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*, pages 521–530. ACM, 2007.

[33] Neil Daswani, Chris Mysen, Vinay Rao, Stephen Weis, Kourosh Gharachorloo, and Shuman Ghosemajumder. Online advertising fraud. *Crimeware: understanding new attacks and defenses*, 2008.

[34] Luis Miralles Pechuán, Enrique Muñoz Ballester, and Jose Manuel García Carrasco. Online advertising and the cpa model: Challenges and opportunities. *International Journal of Engineering and Management Research*, 4(3):324–334, 2014.

[35] Benjamin Edelman. *Deterring online advertising fraud through optimal payment in arrears*. Springer, 2009.

[36] Kenneth C Wilbur and Yi Zhu. Click fraud. *Marketing Science*, 28(2):293–308, 2009.

[37] Hila Becker, Andrei Broder, Evgeniy Gabrilovich, Vanja Josifovski, and Bo Pang. What happens after an ad click?: quantifying the impact of landing pages in web advertising. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 57–66. ACM, 2009.

[38] Benjamin Edelman and Wesley Brandi. Risk, information, and incentives in online affiliate marketing. *Journal of Marketing Research*, 52(1):1–12, 2015.

[39] Michaela Draganska, Wesley R Hartmann, and Gena Stanglein. Internet versus television advertising: A brand-building comparison. *Journal of Marketing Research*, 51(5):578–590, 2014.

[40] Brett Stone-Gross, Ryan Stevens, Apostolis Zarras, Richard Kemmerer, Chris Kruegel, and Giovanni Vigna. Understanding fraudulent activities in online ad exchanges. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 279–294. ACM, 2011.

[41] Kevin Springborn and Paul Barford. Impression fraud in on-line advertising via pay-per-view networks. In *USENIX Security*, pages 211–226, 2013.

[42] Linfeng Zhang and Yong Guan. Detecting click fraud in pay-per-click streams of online advertising networks. In *The 28th International Conference on Distributed Computing Systems, 2008.*, pages 77–84. IEEE, 2008.

[43] Hamid Nazerzadeh, Amin Saberi, and Rakesh Vohra. Dynamic cost-per-action mechanisms and applications to online advertising. In *Proceedings of the 17th international conference on World Wide Web*, pages 179–188. ACM, 2008.

[44] G Kirubavathi and R Anitha. Botnets: A study and analysis. In *Computational Intelligence, Cyber Security and Computational Models*, pages 203–214. Springer, 2014.

[45] Ahmad Karim, Rosli Bin Salleh, Muhammad Shiraz, Syed Adeel Ali Shah, Irfan Awan, and Nor Badrul Anuar. Botnet detection techniques: review, future trends, and issues. *Journal of Zhejiang University SCIENCE C*, 15(11):943–983, 2014.

[46] Charles C Mann. How click fraud could swallow the internet. *Wired Magazine*, pages 17–20, 2006.

[47] N Vidyasagar. India's secret army of online ad clickers. *The Times of India*, 3:2004, 2004.

[48] Brad Miller, Paul Pearce, Chris Grier, Christian Kreibich, and Vern Paxson. What's clicking what? techniques and innovations of today's clickbots. In *Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 164–183. Springer, 2011.

[49] Kenneth Fjell. Online advertising: Pay-per-view versus pay-per-click-a comment. *Journal of Revenue & Pricing Management*, 8(2):200–206, 2009.

[50] Jill Zucker and Tamar R Shapiro. Systems and methods for optimizing marketing decisions based on visitor profitability, January14 2015. US Patent App. 14/596,974.

[51] Sergio Duarte Torres, Ingmar Weber, and Djoerd Hiemstra. Analysis of search and browsing behavior of young users on the web. *ACM Transactions on the Web (TWEB)*, 8(2):7, 2014.

[52] Rohit Kumar, Sneha Manjunath Naik, Vani D Naik, Smita Shiralli, VG Sunil, and Moula Husain. Predicting clicks: CTR estimation of advertisements using logistic regression classifier. In *Advance Computing Conference (IACC), 2015 IEEE International*, pages 1134–1138. IEEE, 2015.

[53] Jongwon Lee, Yong Shi, Fang Wang, Heeseok Lee, and Heung Kee Kim. Advertisement clicking prediction by using multiple criteria mathematical programming. *World Wide Web*, pages 1–18, 2015.

[54] Thore Graepel, Joaquin Q Candela, Thomas Borchert, and Ralf Herbrich. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 13–20, 2010.

[55] Mihajlo Grbovic, Nemanja Djuric, Vladan Radosavljevic, and Narayan Bhamidipati. Search retargeting using directed query embeddings. In *Proceedings of the 24th International Conference on World Wide Web Companion*, pages 37–38. International World Wide Web Conferences Steering Committee, 2015.

[56] Deepak Agarwal, Bee-Chung Chen, and Pradheep Elango. Spatio-temporal models for estimating click-through rate. In *Proceedings of the 18th international conference on World wide web*, pages 21–30. ACM, 2009.

[57] Ameet Ranadive, Shariq Rizvi, and Neilkumar Murli Daswani. Malicious advertisement detection and remediation, August 20 2013. US Patent 8,516,590.

[58] D Vasumati, M Sree Vani, R Bhramaramba, and O Yaswanth Babu. Data mining approach to filter click-spam in mobile ad networks. 2015.

[59] Sneha Singh and Sandeep Kaur. Improved spambase dataset prediction using svm rbf kernel with adaptive boost. 2015.

[60] Samaneh Beheshti-Kashi, Hamid Reza Karimi, Klaus-Dieter Thoben, Michael Lütjen, and Michael Teucke. A survey on retail sales forecasting and prediction in fashion markets. *Systems Science & Control Engineering: An Open Access Journal*, 3(1):154–161, 2015.

[61] Hamed Valizeh Haghi and SM Moghaddas Tafreshi. An overview and verification of electricity price forecasting models. In *Power Engineering Conference, 2007. IPEC 2007. International*, pages 724–729. IEEE, 2007.

[62] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1-2):5–43, 2003.

[63] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. Advances in knowledge discovery and data mining. chapter From Data Mining to Knowledge Discovery: An Overview, pages 1–34. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996.

[64] Oded Maimon and Lior Rokach. *Data mining and knowledge discovery handbook*, volume 2. Springer, 2005.

[65] Michael J Berry and Gordon Linoff. *Data mining techniques: for marketing, sales, and customer support*. John Wiley & Sons, Inc., 1997.

[66] Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229, 1959.

[67] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.

[68] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[69] Ian H Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[70] Machine learning, neural and statistical classification. 1994.

[71] Angela P Ansuj, ME Camargo, R Radharamanan, and DG Petry. Sales forecasting using time series and neural networks. *Computers & Industrial Engineering*, 31(1):421–424, 1996.

[72] Indranil Bose and Radha K Mahapatra. Business data mining - a machine learning perspective. *Information & management*, 39(3):211–225, 2001.

[73] Marco Hülsmann, Detlef Borscheid, Christoph M Friedrich, and Dirk Reith. General sales forecast models for automobile markets and their analysis. *Trans. MLDM*, 5(2):65–86, 2012.

[74] Suresh Kumar Sharma and Vinod Sharma. Comparative analysis of machine learning techniques in sale forecasting. *International Journal of Computer Applications*, 53(6), 2012.

[75] Atul B Borade and Satish V Bansod. Comparison of neural network-based forecasting methods using multi-criteria decision-making tools. In *Supply Chain Forum: An International Journal*, volume 12, pages 4–14. Taylor & Francis, 2011.

[76] Mehdi Khashei, Seyed Reza Hejazi, and Mehdi Bijari. A new hybrid artificial neural networks and fuzzy regression model for time series forecasting. *Fuzzy Sets and Systems*, 159(7):769–786, 2008.

[77] S. B. Kotsiantis. Supervised machine learning: A review of classification techniques, 2007.

[78] Nicole Immorlica, Kamal Jain, Mohammad Mahdian, and Kunal Talwar. Click fraud resistant methods for learning click-through rates. In *Internet and Network Economics*, pages 34–45. Springer, 2005.

[79] Liyang Wei, Yongyi Yang, Robert M Nishikawa, and Yulei Jiang. A study on several machine-learning methods for classification of malignant and benign clustered microcalcifications. *IEEE Transactions on Medical Imaging*, 24(3):371–380, 2005.

[80] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, 42(1-2):177–196, 2001.

[81] Douglas H Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine learning*, 2(2):139–172, 1987.

[82] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[83] Yoshua Bengio. Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[84] Abdel-Rahman Mohamed, Tara N Sainath, George Dahl, Bhuvana Ramabhadran, Geoffrey E Hinton, Michael Picheny, et al. Deep belief networks using discriminative features for phone recognition. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5060–5063. IEEE, 2011.

[85] Sheng-hua Zhong, Yan Liu, and Yang Liu. Bilinear deep learning for image classification. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 343–352. ACM, 2011.

[86] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[87] Quoc V Le. Building high-level features using large scale unsupervised learning. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8595–8598. IEEE, 2013.

[88] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616. ACM, 2009.

[89] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[90] Graham W Taylor, Geoffrey E Hinton, and Sam T Roweis. Modeling human motion using binary latent variables. In *Advances in neural information processing systems*, pages 1345–1352, 2006.

[91] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on AI (Ijcai)*, volume 14, pages 1137–1145, 1995.

[92] Thiago S Guzella and Walmir M Caminhas. A review of machine learning approaches to spam filtering. *Expert Systems with Applications*, 36(7):10206–10222, 2009.

[93] David Heckerman, Eric Horvitz, Mehran Sahami, and Susan Dumais. A bayesian approach to filtering junk e-mail. In *Proceeding of AAAI-98 Workshop on Learning for Text Categorization*, pages 55–62, 1998.

[94] James Clark, Irena Koprinska, and Josiah Poon. A neural network based approach to automated e-mail classification. In *null*, page 702. IEEE, 2003.

[95] Joshua Goodman and Wen-tau Yih. Online discriminative spam filter training. In *Conference on Email and Anti-Spam (CEAS)*, pages 1–4, 2006.

[96] Konstantin Tretyakov. Machine learning techniques in spam filtering. In *Data Mining Problem-oriented Seminar, MTAT*, volume 3, pages 60–79. Citeseer, 2004.

[97] K Bauman, A Kornetova, V Topinsky, and D Leshiner. Ctr prediction based on click statistic. In *Workshop: Machine Learning in Online Advertising*, pages 8–13. Citeseer, 2012.

[98] Suhrid Balakrishnan, Sumit Chopra, and I Dan Melamed. The business next door: Click-through rate modeling for local search. *Machine Learning in Online Advertising*, page 14, 2010.

[99] Zhipeng Fang, Kun Yue, Jixian Zhang, Dehai Zhang, and Weiyi Liu. Predicting click-through rates of new advertisements based on the bayesian network. *Mathematical Problems in Engineering*, 2014, 2014.

[100] Dawei Yin, Shike Mei, Bin Cao, Jian-Tao Sun, and Brian D Davison. Exploiting contextual factors for click modeling in sponsored search. In *Proceedings of the*

7th ACM international conference on Web search and data mining, pages 113–122. ACM, 2014.

[101] Gouthami Kondakindi, Satakshi Rana, Aswin Rajkumar, Sai Kaushik Ponnekanti, and Vinit Parakh. A logistic regression approach to ad click prediction. *Machine Learning-Class Project*, 2014.

[102] Yukihiro Tagami, Shingo Ono, Koji Yamamoto, Koji Tsukamoto, and Akira Tajima. Ctr prediction for contextual advertising: Learning-to-rank approach. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, page 4. ACM, 2013.

[103] Hema Yoganarasimhan. Search personalization using machine learning. *Available at SSRN 2590020*, 2015.

[104] Krzysztof Dembczynski, Wojciech Kotlowski, and Dawid Weiss. Predicting ads click-through rate with decision rules. In *Workshop on targeting and ranking in online advertising*, 2008.

[105] Ilya Trofimov, Anna Kornetova, and Valery Topinskiy. Using boosted trees for click-through rate prediction for sponsored search. In *Proceedings of the Sixth International Workshop on Data Mining for Online Advertising and Internet Economy*, page 2. ACM, 2012.

[106] Fei-Long Chen and Tsung-Yin Ou. Constructing a sales forecasting model by integrating gra and elm: A case study for retail industry. *International Journal of Electronic BusinessManagement*, 9(2):107, 2011.

[107] Huan Liu and Hiroshi Motoda. *Feature extraction, construction and selection: A data mining perspective*. Springer Science & Business Media, 1998.

[108] Vipin Kumar and Sonajharia Minz. Feature selection. *SmartCR*, 4(3):211–229, 2014.

[109] Rich Caruana and Dayne Freitag. Greedy attribute selection. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 28–36. Morgan Kaufmann, 1994.

[110] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

[111] Abbas Khalili. An overview of the new feature selection methods in finite mixture of regression models. *Journal of Iranian Statistical Society*, 10(2):201–235, 2011.

[112] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

[113] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[114] Jasmina Novakovic. Using information gain attribute evaluation to classify sonar targets. In *17th Telecommunications forum TELFOR*, pages 24–26, 2009.

[115] Asha Gowda Karegowda, AS Manjunath, and MA Jayaram. Comparative study of attribute selection using gain ratio and correlation based feature selection. *International Journal of Information Technology and Knowledge Management*, 2(2):271–277, 2010.

[116] Igor Kononenko. Evaluating the quality of attributes. *Advanced Course on Knowledge Technologies, ACAI*, 2005.

[117] Jasmina Novaković, Perica Štrbac, and Dušan Bulatović. Toward optimal feature selection using ranking methods and classification algorithms. *Yugoslav Journal of Operations Research ISSN: 0354-0243 EISSN: 2334-6043*, 21(1), 2011.

[118] Ji-Hyun Kim. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational Statistics & Data Analysis*, 53(11):3735–3745, 2009.

[119] W. Siedlecki and J. Sklansky. A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, 10(5):335 – 347, 1989.

[120] Haleh Vafaie and Kenneth De Jong. Genetic algorithms as a tool for feature selection in machine learning. pages 200–204. Society Press, 1992.

[121] M.E. ElAlami. A filter model for feature subset selection based on genetic algorithm. *Knowledge-Based Systems*, 22(5):356 – 362, 2009.

[122] R.C. Anirudha, R. Kannan, and N. Patil. Genetic algorithm based wrapper feature selection on hybrid prediction model for analysis of high dimensional data. In *9th International Conference on Industrial and Information Systems (ICIIS), 2014*, pages 1–6, Dec 2014.

[123] Jinjie Huang, Yunze Cai, and Xiaoming Xu. A hybrid genetic algorithm for feature selection wrapper based on mutual information. *Pattern Recognition Letters*, 28(13):1825 – 1844, 2007.

[124] Antonio F. Gómez-Skarmeta, Fernando Jiménez, Jesús Ibánez, and Santiago Paredes. Evolutionary variable identification. In *7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99)*, 1999.

[125] J. Yang and V. Honavar. Feature subset selection using a genetic algorithm. *Intelligent Systems and their Applications, IEEE*, 13(2):44–49, Mar 1998.

[126] Sigve Dreyer. Evolutionary feature selection. 2013.

[127] C.A. Coello, Veldhuizen D.V., and Lamont G.B. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic/Plenum publishers, New York, NY, USA, 2002.

[128] K. Deb. *Multi-objective optimization using evolutionary algorithms*. Wiley, London, UK, 2001.

[129] Hisao Ishibuchi. Multi-objective pattern and feature selection by a genetic algorithm. In *Proc. of Genetic and Evolutionary Computation Conference (GECCO'2000),*, pages 1069–1076. Morgan Kaufmann, 2000.

[130] C. Emmanouilidis, A. Hunter, J. MacIntyre, and C. Cox. A multi-objective genetic algorithm approach to feature selection in neural and fuzzy modeling. *Journal of Evolutionary Optimization, An International Journal on the Internet*, 3(1):1–26, 2001.

[131] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.

[132] O. Cordón, F. Herrera, M.J. del Jesús, and P. Villar. A multiobjective genetic algorithm for feature selection and granularity learning in fyzzy-rule based classifiction systems. In *Proceedings of the IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*, volume 3, pages 1253–1258, 20021.

[133] Carlos M Fonseca, Peter J Fleming, et al. Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization., 1993.

[134] Juan Liu and Hitoshi Iba. Selecting informative genes using a multiobjective evolutionary algorithm. In *Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC'02.*, volume 1, pages 297–302. IEEE, 2002.

[135] Gisele Pappa, Alex Freitas, and Celso Kaestner. Attribute selection with a multiobjective genetic algorithm. In Guilherme Bittencourt and GeberL. Ramalho, editors, *Advances in Artificial Intelligence*, volume 2507 of *Lecture Notes in Computer Science*, pages 280–290. Springer Berlin Heidelberg, 2002.

[136] S.Y.M. Shi, P.N. Suganthan, and K. Deb. Multiclass protein fold recognition using multiobjective evolutionary algorithms. In *Proceedings of the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, 2004. CIBCB '04.*, pages 61–66, Oct 2004.

[137] K. Deb, A. Pratab, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182 – 197, 2002.

[138] Huanhuan Chen and Xin Yao. Evolutionary multiobjective ensemble learning based on bayesian feature selection. In *IEEE Congress on Evolutionary Computation, 2006. CEC 2006.*, pages 267–274, 2006.

[139] Zexuan Zhu, Yew-Soon Ong, and Jer-Lai Kuo. Feature selection using single/multi-objective memetic frameworks. In Chi-Keong Goh, Yew-Soon Ong, and KayChen Tan, editors, *Multi-Objective Memetic Algorithms*, volume 171 of *Studies in Computational Intelligence*, pages 111–131. Springer Berlin Heidelberg, 2009.

[140] M. Venkatadri and K. Srinivasa Rao. A multiobjective genetic algorithm for feature selection in data mining. *International Journal of Computer Science and Information Technologies*, 1(5):443–448, 2010.

[141] A. Ekbal, S. Saha, and C.S. Garbe. Feature selection using multiobjective optimization for named entity recognition. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 1937–1940, Aug 2010.

[142] Alan P Reynolds, David W Corne, and Michael J Chantler. Feature selection for multi-purpose predictive models: a many-objective task. In *Parallel Problem Solving from Nature, PPSN XI*, pages 384–393. Springer, 2010.

[143] António Gaspar-Cunha. Feature selection using multi-objective evolutionary algorithms: Application to cardiac spect diagnosis. In Miguel P. Rocha, Florentino Fernández Riverola, Hagit Shatkay, and Juan Manuel Corchado, editors, *Advances in Bioinformatics: 4th International Workshop on Practical Applications of Computational Biology and Bioinformatics 2010 (IWPACBB 2010)*, pages 85–92. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[144] António Gaspar-Cunha and José A Covas. Rpsgae-reduced pareto set genetic algorithm: application to polymer extrusion. In *Metaheuristics for Multiobjective Optimisation*, pages 221–249. Springer, 2004.

[145] Pablo AD Castro and Fernando J Von Zuben. Multi-objective feature selection using a bayesian artificial immune system. *International Journal of Intelligent Computing and Cybernetics*, 3(2):235–256, 2010.

[146] Igor Vatolkin, Mike Preuß, and Günter Rudolph. Multi-objective feature selection in music genre and style recognition tasks. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, pages 411–418, New York, NY, USA, 2011. ACM.

[147] A. Jara, R. Martínez, D. Vigueras, G. Sánchez, and F. Jiménez. Attribute selection by multiobjective evolutionary computation applied to mortality from infection in severe burns patients. In *HEALTHINF 2011 - Proceedings of the International Conference on Health Informatics, Rome, Italy, 26-29 January, 2011*, pages 467–471, 2011.

[148] Brahmadesam Krishna and Baskaran Kaliaperumal. Efficient genetic-wrapper algorithm based data mining for feature subset selection in a power quality pattern recognition application. *Int. Arab J. Inf. Technol.*, 8(4):397 – 405, 2011.

[149] Hossein Karshenas, Pedro Larrañaga Múgica, Qingfu Zhang, and Concha Bielza. An interval-based multiobjective approach to feature subset selection using joint modeling of objectives and variables. 2012.

[150] Soumen Kumar Pati, Asit Kumar Das, and Arka Ghosh. Gene selection using multiobjective genetic algorithm integrating cellular automata and rough set theory. In *International Conference on Swarm, Evolutionary, and Memetic Computing*, pages 144–155. 2013.

[151] Dragi Kimovski, Julio Ortega, Andrés Ortiz, and Raúl Baños. Parallel alternatives for evolutionary multi-objective optimization in unsupervised feature selection. *Expert Systems with Applications*, 42(9):4239–4252, 2015.

[152] F. Jiménez, A.F. Gómez-Skarmeta, G. Sánchez, and K. Deb. An evolutionary algorithm for constrained multi-objective optimization. In *Proceedings of the Evolutionary Computation on 2002. CEC '02.*, volume 2 of *CEC '02*, pages 1133–1138, Washington, DC, USA, 2002. IEEE Computer Society.

[153] Fernando Jiménez, Gracia Sánchez, and José M. Juárez. Multi-objective evolutionary algorithms for fuzzy classification in survival prediction. *Artificial Intelligence in Medicine*, 60(3):197–219, 2014.

[154] I. Rechenberg. *Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog, Stuttgart, Germany, 1973.

[155] Hans-Paul Schwefel. *Numerical optimization of computer models*. John Wiley & Sons, Inc., 1981.

[156] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, volume 2(12), pages 1137–1143. Morgan Kaufmann, 1995.

[157] Vipin Kumar and Sonajharia Minz. Feature selection: A literature review. *Smart CR*, 4(3):211–229, 2014.

[158] David E Goldberg and John H Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988.

[159] Richard K. Belew, John Mcinerney, and Nicol N. Schraudolph. Evolving networks: Using the genetic algorithm with connectionist learning. In *In*, pages 511–547. Addison-Wesley, 1990.

[160] Daniel S Weile and Eric Michielssen. Genetic algorithm optimization applied to electromagnetics: A review. *IEEE Transactions on Antennas and Propagation,*, 45(3):343–353, 1997.

[161] Darrell Whitley. A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85, 1994.

[162] Min Chen, Varghese S Jacob, Suresh Radhakrishnan, and Young U Ryu. The effect of fraud investigation cost on pay-per-click advertising. In *WEIS*, 2012.

[163] Tamara Dinev, Qing Hu, and Ali Yayla. Is there an on-line advertisers' dilemma? a study of click fraud in the pay-per-click model. *International Journal of Electronic Commerce*, 13(2):29–60, 2008.

[164] Wendy Hui. Estimating the number of genuine and fraudulent clicks in the pay-per-click (ppc) model. *Available at SSRN 1544173*, 2010.

[165] Bob Mungamuru, Stephen Weis, and Hector Garcia-Molina. Should ad networks bother fighting click fraud? (yes, they should.). (2008-24), July 2008.

[166] Richard Chow, Philippe Golle, Markus Jakobsson, Lusha Wang, and XiaoFeng Wang. Making captchas clickable. In *Proceedings of the 9th workshop on Mobile computing systems and applications*, pages 91–94. ACM, 2008.

[167] Hamed Haddadi. Fighting online click-fraud using bluff ads. *ACM SIGCOMM Computer Communication Review*, 40(2):21–25, 2010.

[168] GM Adelson-Velskii and Evgenii Mikhailovich Landis. An information organization algorithm. In *Doklady Akademia Nauk SSSR*, volume 146, pages 263–266, 1962.

[169] Claudia Perlich, Brian Dalessandro, Troy Raeder, Ori Stitelman, and F Provost. Machine learning for targeted display advertising: Transfer learning in action. *Machine learning*, 95(1):103–127, 2014.

[170] Avi Goldfarb and Catherine E Tucker. Privacy regulation and online advertising. *Management Science*, 57(1):57–71, 2011.

[171] Tommy Blizard and Nikola Livic. Click-fraud monetizing malware: A survey and case study. In *7th International Conference on Malicious and Unwanted Software (MALWARE), 2012*, pages 67–72. IEEE, 2012.

[172] Nitish Korula, Vahab Mirrokni, and Hamid Nazerzadeh. Optimizing display advertising markets: Challenges and directions. *IEEE Internet Computing*, 20(1):28–35, January 2016.

[173] Moneet Singh. Fraud detection for use in payment processing, December 12 2006. US Patent App. 11/638,290.

[174] Apostolis Zarras, Alexandros Kapravelos, Gianluca Stringhini, Thorsten Holz, Christopher Kruegel, and Giovanni Vigna. The dark alleys of madison avenue: Understanding malicious advertisements. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 373–380. ACM, 2014.

[175] Richard Oentaryo, Ee-Peng Lim, Michael Finegold, David Lo, Feida Zhu, Clifton Phua, Eng-Yeow Cheu, Ghim-Eng Yap, Kelvin Sim, Minh Nhut Nguyen, et al. Detecting click fraud in online advertising: a data mining approach. *The Journal of Machine Learning Research*, 15(1):99–140, 2014.

[176] Beatriz Plaza. Google analytics for measuring website performance. *Tourism Management*, 32(3):477–481, 2011.

[177] Anindya Ghose and Sha Yang. An empirical analysis of search engine advertising: Sponsored search in electronic markets. *Management Science*, 55(10):1605–1622, 2009.

[178] Ying Cui, Ruofei Zhang, Wei Li, and Jianchang Mao. Bid landscape forecasting in online ad exchange marketplace. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 265–273. ACM, 2011.

[179] Google AdSense. Google adsense - terms and conditions. `https://www.google.com/adsense/localized-terms?hl=en_US`, 2015.

[180] Nir Kshetri. The economics of click fraud. *IEEE Security & Privacy*, 8(3):45–53, 2010.

[181] John Newcombe. Genetic algorithm framework - john newcombe. `http://johnnewcombe.net/gaf`, December 2015. [Online; accessed 05-December-2015].

[182] Eric Bax, Anand Kuratti, Preston Mcafee, and Julian Romero. Comparing predicted prices in auctions for online advertising. *International Journal of Industrial Organization*, 30(1):80–88, 2012.

[183] Chang-Hoan Cho. Why do people avoid advertising on the internet? *Journal of advertising*, 33(4):89–97, 2004.

[184] Ritu Lohtia, Naveen Donthu, and Edmund K Hershberger. The impact of content and design elements on banner advertising click-through rates. *Journal of advertising Research*, 43(04):410–418, 2003.

[185] Benjamin Rey and Ashvin Kannan. Conversion rate based bid adjustment for sponsored search. In *Proceedings of the 19th international conference on World wide web*, pages 1173–1174. ACM, 2010.

[186] Yu Jeffrey Hu, Jiwoong Shin, and Zhulei Tang. Pricing of online advertising: Cost-per-click-through vs. cost-per-action. In *43rd Hawaii International Conference on System Sciences (HICSS), 2010*, pages 1–9. IEEE, 2010.

[187] Evgeniy Gabrilovich, Andrei Broder, Marcus Fontoura, Amruta Joshi, Vanja Josifovski, Lance Riedel, and Tong Zhang. Classifying search queries using the web as a source of knowledge. *ACM Transactions on the Web (TWEB)*, 3(2):5, 2009.

[188] The caret Package. The caret package (short for Classification And REgression Training). `http://topepo.github.io/caret/index.html`, July 2015. [Online; accessed 05-July-2015].

[189] R Studio. RStudio is free and open source data analysis). `https://www.rstudio.com/`, June 2015. [Online; accessed 23-June-2015].

[190] Max Kuhn and Kjell Johnson. *Applied predictive modeling*. Springer, 2013.

[191] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1):1–21, 2015.

[192] Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, pages 473–480. ACM, 2007.

[193] Ramón Díaz-Uriarte and Sara Alvarez De Andres. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1):3, 2006.

[194] Andrea Mangani. Online advertising: Pay-per-view versus pay-per-click. *Journal of Revenue and Pricing Management*, 2(4):295–302, 2004.

[195] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-validation. In *Encyclopedia of database systems*, pages 532–538. Springer, 2009.

[196] Mona Gandhi, Markus Jakobsson, and Jacob Ratkiewicz. Badvertisements: Stealthy click-fraud with unwitting accessories. *Journal of Digital Forensic Practice*, 1(2):131–142, 2006.

[197] Maryam Feily, Alireza Shahrestani, and Sureswaran Ramadass. A survey of botnet and botnet detection. In *Third International Conference on Emerging Security Information, Systems and Technologies, 2009. SECURWARE'09.*, pages 268–273. IEEE, 2009.

[198] Andrew F Tappenden and James Miller. Cookies: A deployment study and the testing implications. *ACM Transactions on the Web (TWEB)*, 3(3):9, 2009.

[199] Max Kuhn. Building predictive models in r using the caret package. *Journal of Statistical Software*, 28(5):1–26, 2008.

[200] H2O R Studio Installation. H2O installation in R Studio - H2O 2.3.0.1283 documentation. `http://docs.h2o.ai/h2oclassic/Ruser/Rinstall.html`, April 2015. [Online; accessed 8-April-2015].

[201] Weka. Weka 3: Data mining software in java. `http://www.cs.waikato.ac.nz/ml/weka/`, June 2015. [Online; accessed 11-June-2015].

[202] Deep Learning H2O Documentation. Deep Learning - H2O 2.8.6.2 documentation. `https://s3.amazonaws.com/h2o-release/h2o/rel-markov/1/docs-website/datascience/deeplearning.html`, April 2015. [Online; accessed 22-April-2015].

[203] Max Kuhn. Variable selection using the caret package. 2012.

[204] Internet Advertisements Data Set. UCI Machine Learning Repository: Internet Advertisements. `https://archive.ics.uci.edu/ml/datasets/Internet+Advertisements`, Jun 2015. [Online; accessed 16-June-2015].

[205] Kaggle: The Home of Data Science. Data - Display Advertising Challenge | Kaggle. `www.kaggle.com/c/criteo-display-ad-challenge/data`, July 2015. [Online; accessed 16-July-2008].

[206] Online Product Sales. Description - Online Product Sales | Kaggle. `https://www.kaggle.com/c/online-sales`, July 2015. [Online; accessed 22-July-2015].

[207] Li Deng and Dong Yu. Deep learning: methods and applications. *Foundations and Trends in Signal Processing*, 7(3–4):197–387, 2014.

[208] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[209] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[210] M. Srinivas and L.M. Patnaik. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(4):656–667, 1994.

[211] M Kuhn, Jed Wing, Stew Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, and Zachary Mayer. Caret: Classification and regression training. r package version 6.0-21. `https://cran.r-project.org/web/packages/caret/`, 2014.

[212] Weka. Weka 3: Data mining software in java. http://www.cs.waikato.ac.nz/ml/weka/, June 2015. [Online; accessed 11-June-2015].

[213] Donald W Zimmerman and Bruno D Zumbo. Relative power of the wilcoxon test, the friedman test, and repeated-measures anova on ranks. *The Journal of Experimental Education*, 62(1):75–86, 1993.