(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 March 2019 (21.03.2019)

WIPO | PCT

(10) International Publication Number
WO 2019/052869 A1

(51) International Patent Classification:
G06Q 30/02 (2012.01) *G06N 5/00* (2006.01)

(21) International Application Number:
PCT/EP2018/073841

(22) International Filing Date:
05 September 2018 (05.09.2018)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
1758514 14 September 2017 (14.09.2017) FR
15/704,320 14 September 2017 (14.09.2017) US

(71) Applicant: **AMADEUS S.A.S.** [FR/FR]; 485 route du Pin Montard, Sophia Antipolis, 06410 Biot (FR).

(72) Inventors: **ACUNA AGOST, Rodrigo**; 652 Chemin de la Mignonnette, 06220 Golfe Juan (FR). **MOTTINI D'OLIVEIRA, Alejandro Ricardo**; 6 rue de la boucherie, 06300 Nice (FR). **RENAUDIE, David**; 806 Chemin de Peidessalle, Batiment Chevrefeuille, 06560 Valbonne (FR).

(74) Agent: **SAMSON & PARTNER PATENTANWÄLTE MBB, ASSOCIATION NO. 275**; Widenmayerstrasse 6, 80538 München (DE).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report (Art. 21(3))*

(54) Title: A MACHINE LEARNING METHOD AND SYSTEM FOR PREDICTING ONLINE USER INTERACTIONS

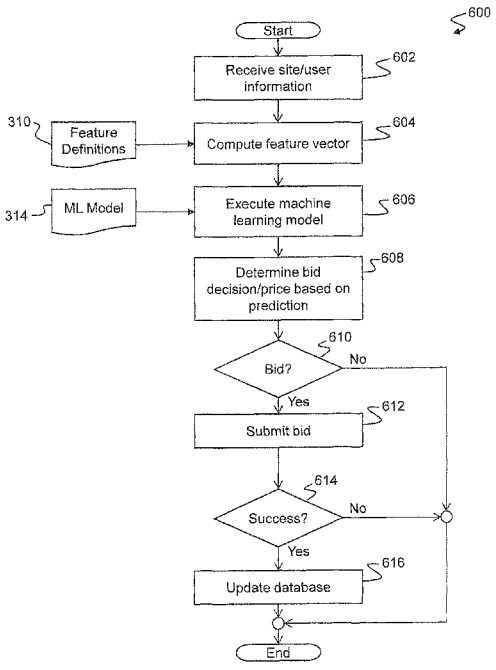

Figure 6

(57) Abstract: A computer-implemented method comprises accessing an online data store to retrieve records relating to content placement events, and records relating to user interaction events. A set of enriched training feature vectors is computed from raw feature values, and used with interaction event tags to train a machine learning model. A processor is configured to execute the machine learning model, and receives information relating to an online content placement slot and information relating to a user. The processor computes an enriched estimation feature vector based upon a content item selected for placement within the online content placement slot, the information relating to the user, and the information relating to the online content placement slot. The processor executes the machine learning model to determine an estimate of likelihood of the user interacting with the selected content item, based upon the enriched estimation feature vector.

WO 2019/052869 A1

A MACHINE LEARNING METHOD AND SYSTEM FOR PREDICTING ONLINE USER INTERACTIONS

FIELD OF THE INVENTION

[0001] The present invention relates to the application of machine learning models for predicting behaviour of online users. In particular, embodiments of the invention predict likelihood of user interaction with online content elements based upon aggregated behaviour of prior users in similar contexts. The invention may be applied in online advertising systems, for example to determine whether or not to bid for placement of an advertisement to be presented to a user, e.g. via a web page on within a mobile app.

BACKGROUND TO THE INVENTION

[0002] Online (e.g. web-based, mobile, or in-app) advertising differs from advertising in traditional media in its degree of personalised audience targeting. For example, while broadcast media advertising, such as television advertising, aims to reach a target demographic defined by broad characteristics such as age-group, socioeconomic status, and/or general interests, online advertising aims to reach individuals having a particular interest in the product, service, or information that is presented.

[0003] Highly personalised audience targeting technology has led to the development of business models that are specific to online advertising. For example, it is now common for websites that provide news, aggregated information, and other content of interest to particular users, to host third-party advertisements as a means for generating revenue. Advertisers whose advertisements appear on these websites may pay the operator on the basis of viewing opportunities or impressions (commonly measured as 'cost per thousand impressions', a.k.a. CPM), on the basis of a cost per click (CPC), or according to some other measure of performance. The actual selection of an advertisement to

be placed on a web page presented to an individual user may be based, at least in part, on a bidding process whereby an advertiser who is willing to pay a higher CPM, CPC, or other cost measure, is more likely to have its advertisement presented to the user.

[0004] According to one common model, the bidding process is facilitated by an 'ad exchange platform'. An ad exchange is a technology platform that implements a digital marketplace allowing advertisers and publishers of web sites and other online content to buy and sell advertising space, often through real-time auctions. Well-known ad exchange platforms include DoubleClick™ (owned by Google™), AppNexus™, Microsoft™ Ad Exchange™, and OpenX™.

[0005] An ad exchange maintains a 'pool' of ad slots. Publishers contribute their ad slots, e.g. available advertising slots embedded within web pages served to users, into the pool. Buyers can then bid for the slots that they wish to purchase. Bidding decisions are often made in real time based on information such as the previous behaviour of the user an ad is being served to, time of day, device type, ad position, and so forth. In practice, these bidding decisions must themselves be made very rapidly, e.g. in at most a few tens of milliseconds, using technology platforms commonly known as demand side platforms (DSPs). Since there is a real cost to the advertiser in purchasing impressions through an ad exchange, the performance of technologies and algorithms deployed in a DSP for assessing the potential 'value' of a user in order to make a bid decision may have a significant business impact.

[0006] By way of example, an average click through rate (CTR) for web-based display ads is around 0.05%, i.e. five clicks per 10,000 impressions. Where ad slots are purchased via an ad exchange, every impression represents a cost (i.e. the price paid following a winning bid). However, where a CPC model is employed, only 0.05% of impressions, on average, result in the generation of revenue. A low CTR thus results in an inefficient use of technical resources (such as the processing resources of the DSP) and a higher cost to successful

advertisers (since a smaller number of 'clicks' must cover the cost of all impressions). Furthermore, a low CTR is indicative of a lack of suitability or relevance of the displayed advertising to online users.

[0007] It is, therefore, highly desirable to deploy technologies in DSP platforms that result in higher CTR. Such technologies benefit DSP operators by enhancing the utilisation of technical resources and increasing revenues generated from user interactions with advertising content. Advertisers also benefit from more effective and successful placement of ads, resulting in greater click-though, and delivery of a larger audience of potential interested customers. Online users also benefit, since higher CTR is generally achieved by placement of ads having greater suitability and relevance to users.

[0008] One common approach to increasing CTR employs online tracking to infer user interests and to place advertising content that is relevant to those interests. For example, browser cookies and other tracking technologies may be used to gather information regarding, for example, web sites visited by a user, topics viewed on social media sites, and/or online searches conducted by the user. Such information may be processed to identify advertising content that is inferred to be relevant to the user. However, predicting the likelihood that the user will interact with such content is more difficult. As noted above, average CTR is low, and individual online users therefore rarely interact with advertising content, even when it has been effectively targeted based on user interests. There is, accordingly, very little direct data that can be used to predict the likelihood of users interacting with advertising content on the basis of individual user tracking.

[0009] As a result, there is a clear need for real-time computer-implemented technologies, methods and systems that can be deployed within DSPs and that are able to make improved decisions regarding ad selection and bidding through ad exchange platforms. In particular, it would be desirable to provide improved methods and systems for predicting the likelihood that an online user will interact

with advertising content. In order to meet technical requirements, and ensure that page load times are not unduly increased by the selection and bidding process, it is necessary that such a prediction be made in no more than a few tens of milliseconds. The present invention is directed to addressing these needs.

SUMMARY OF THE INVENTION

[0010] In one aspect, the present invention provides a computer-implemented method comprising:

accessing an online data store to retrieve records relating to content placement events, and records relating to user interaction events, wherein the placement and interaction events occur within a defined time period;

matching retrieved content placement event records with retrieved interaction event records to generate a matched data set which comprises a plurality of records, each record of the matched data set including a set of raw feature values derived from a content placement event along with an interaction event tag indicating whether or not an interaction event occurred corresponding with the content placement event;

computing, from the raw feature values, a corresponding set of enriched training feature vectors;

training a machine learning model using the enriched training feature vectors and corresponding interaction event tags;

receiving, at a processor configured to execute the machine learning model, information relating to an online content placement slot and information relating to a user to whom content within the online content placement slot will be displayed;

computing, by the processor, an enriched estimation feature vector based upon a content item selected for placement within the online content placement slot, the information relating to the user, and the information relating to the online content placement slot;

determining, by the processor executing the machine learning model,

an estimate of likelihood of the user interacting with the selected content item, based upon the enriched estimation feature vector.

[0011] Advantageously, embodiments of the invention employ matching of aggregated content placement events with aggregated user interaction events to support prediction of online user interactions with content using a machine learning model. Content items may be, for example, online ads comprising one or more offers, and the user interaction events may comprise interactions such as clicks on specific offers within the ads. Accordingly, in such embodiments:

the online content placement slot is an ad slot;

the information relating to the ad slot and information relating to the user to whom content within the ad slot will be displayed is received along with a bid request message transmitted from an ad exchange server; and

the content item comprises at least one offer for placement within the ad slot.

[0012] The method may further comprise:

transmitting, to the ad exchange server by the processor, a bid response message in reply to the bid request message;

receiving, by the processor from the ad exchange server, a successful bid notification;

updating, by the processor, the online data store with content placement event data relating to placement of the content item;

receiving, by the processor, a notification of a user interaction with the content item; and

updating (226) the online data store (166) with user interaction event data relating to the user interaction with the content item.

[0013] In this way, embodiments of the invention provide for continuous updating of the stored records of content placement events and user interaction events, enabling the machine learning model to be updated with current information regarding user behaviour. To this end, the method may comprise

repeatedly executing the steps of accessing the online data store, matching the retrieved content placement event records with the retrieved interaction event records, computing enriched training feature vectors, and training the machine learning model.

[0014] In another aspect, the invention provides a computing apparatus which implements a demand side platform, the computing apparatus comprising:

a processor;

at least one memory device accessible by the processor; and

a data communications interface operably associated with the processor,

wherein the memory device contains a body of program instructions including a machine learning model which is executable by the processor and configured to determine an estimate of likelihood of user interaction with a content item, the model having been trained using a set of enriched training feature vectors and corresponding interaction event tags derived from a matched data set generated from records relating to content placement events and records relating to user interaction events retrieved from an online data store wherein the placement and interaction events occur within a defined time period,

the body of program instructions further including instructions which, when executed by the processor, cause the computing apparatus to implement a method comprising steps of:

receiving, via the data communications interface, information relating to an online content placement slot and information relating to a user to whom content within the online content placement slot will be displayed;

computing an enriched estimation feature vector based upon a content item selected for placement within the online content placement slot, the information relating to the user, and the information relating to the corresponding online content placement slot; and

executing the machine learning model to determine an estimate of likelihood of the user interacting with the selected content item, based upon the enriched estimation feature vector.

[0015] In embodiments of the invention, the machine learning model is a generalised linear model comprising a plurality of model coefficients, and in particular the machine learning model may be a logistic regression model.

[0016] Advantageously, the plurality of model coefficients may be stored in a dictionary data structure in which each entry is defined by a key and a coefficient value, wherein each key comprises a hashed representation of a concatenation of a feature name and a corresponding feature value. The use of this type of data structure enables a prediction of the likelihood of a user interaction to be computed very rapidly, e.g. within 30 ms. The program instructions may thus cause the computing apparatus to implement the step of executing the machine learning model by:

generating, for each feature value of the enriched estimation feature vector, a corresponding key;

retrieving from the dictionary data structure, for each generated key, a corresponding coefficient value; and

computing, using the enriched estimation feature vector and the retrieved coefficient values, the estimate of likelihood of the user interacting with the selected content item.

[0017] In embodiments in which the online content placement slot is an ad slot, the information relating to the ad slot and information relating to the user to whom content within the ad slot will be displayed is received along with a bid request message transmitted from an ad exchange server, and the content item comprises at least one offer for placement within the ad slot, the body of program instructions may further include instructions which, when executed by the processor, cause the computing apparatus to implement the method comprising further steps of:

transmitting, to the ad exchange server, a bid response message in reply to the bid request message;

responsive to receiving, from the ad exchange server, a successful bid notification, updating the online data store with content placement event data

relating to placement of the content item; and

responsive to receiving a notification of a user interaction with the content item, updating the online data store with user interaction event data relating to the user interaction with the content item.

[0018] In yet another aspect, the invention provides a computing apparatus which implements training of a machine learning model configured to estimate of likelihood of user interaction with content items, the computing apparatus comprising:

a processor;

at least one memory device accessible by the processor; and

a data store accessible by the processor,

wherein the memory device contains a body of program instructions including instructions which, when executed by the processor, cause the computing apparatus to implement a method comprising steps of:

accessing the data store to retrieve records relating to content placement events, and records relating to user interaction events, wherein the placement and interaction events occur within a defined time period;

matching retrieved content placement event records with retrieved interaction event records to generate a matched data set which comprises a plurality of records, each record of the matched data set including a set of raw feature values derived from a content placement event along with an interaction event tag indicating whether or not an interaction event occurred corresponding with the content placement event;

computing, from the raw feature values, a corresponding set of enriched training feature vectors; and

training the machine learning model using the enriched training feature vectors and corresponding interaction event tags.

[0019] In embodiments of the invention, the machine learning model is a logistic regression model comprising a plurality of model coefficients, and the program instructions cause the computing apparatus to implement the step of training the machine learning model using regularised logistic regression with 'follow-the-regularised-leader' (FTRL)-proximal learning.

[0020] The body of program instructions may further include instructions which, when executed by the processor, cause the computing apparatus to implement the method comprising further steps, for each coefficient of the plurality of model coefficients, of:

generating a key comprising a hashed representation of a feature name and a feature value corresponding with the coefficient; and

storing, in a dictionary data structure, a value of the coefficient in association with the key,

whereby the dictionary data structure comprises an efficient encoding of the machine learning model.

[0021] The body of program instructions may include instructions which, when executed by the processor, cause the computing apparatus to repeatedly execute the steps of accessing the online data store, matching the retrieved content placement event records with the retrieved interaction event records, computing (406) enriched training feature vectors, and training the machine learning model, to update the machine learning model.

[0022] In another aspect, the invention provides a computer program comprising program code instructions for executing the steps of the method according to the first aspect when said program is executed on a computer. The program code instructions may, for example, be stored on tangible machine-readable media.

[0023] Further aspects, advantages, and features of embodiments of the invention will be apparent to persons skilled in the relevant arts from the following

description of various embodiments. It will be appreciated, however, that the invention is not limited to the embodiments described, which are provided in order to illustrate the principles of the invention as defined in the foregoing statements and in the appended claims, and to assist skilled persons in putting these principles into practical effect.

BRIEF DESCRIPTION OF THE DRAWINGS

[0024] Embodiments of the invention will now be described with reference to the accompanying drawings, in which like reference numerals refer to like features, and wherein:

Figure 1 is a schematic diagram illustrating an exemplary networked system embodying the invention;

Figure 2 shows a timeline of communications between a user device, a web server, and ad exchange server, and a DSP embodying the invention;

Figure 3 is a block diagram illustrating schematically a number of code modules comprising an online user interaction prediction engine embodying the invention;

Figure 4 shows a flowchart of a method of online updating of a machine learning model embodying the invention;

Figure 5 shows a flowchart of a method of feature engineering and model hyperparameter optimisation according to an embodiment of the invention;

Figure 6 shows a flowchart of a method of operating a real-time bidding module according to an embodiment of the invention; and

Figures 7(a) and 7(b) show charts illustrating performance of a real-time bidding module embodying the invention.

DETAILED DESCRIPTION OF EMBODIMENTS

[0025] Figure 1 is a block diagram illustrating an exemplary networked system 100 including a demand side platform (DSP) server 102, which is configured to implement a method of bidding for placement of advertising content in

accordance with an embodiment of the invention. The DSP server 102 may comprise a computer system having a conventional architecture. In particular, the DSP server 102, as illustrated, comprises a processor 104. The processor 104 is operably associated with a non-volatile memory/storage device 106, e.g. via one or more data/address busses 108 as shown. The non-volatile storage 106 may be a hard disk drive, and/or may include a solid-state non-volatile memory, such as ROM, flash memory, solid-state drive (SSD), or the like. The processor 104 is also interfaced to volatile storage 110, such as RAM, which contains program instructions and transient data relating to the operation of the DSP server 102.

[0026] In a conventional configuration, the storage device 106 maintains known program and data content relevant to the normal operation of the DSP server 102. For example, the storage device 106 may contain operating system programs and data, as well as other executable application software necessary for the intended functions of the authentication server 102. The storage device 106 also contains program instructions which, when executed by the processor 104, cause the DSP server 102 to perform operations relating to an embodiment of the present invention, such as are described in greater detail below, and with reference to Figures 2 and 6 in particular. In operation, instructions and data held on the storage device 106 are transferred to volatile memory 110 for execution on demand.

[0027] The processor 104 is also operably associated with a communications interface 112 in a conventional manner. The communications interface 112 facilitates access to a wide-area data communications network, such as the Internet 116.

[0028] In use, the volatile storage 110 contains a corresponding body 114 of program instructions transferred from the storage device 106 and configured to perform processing and other operations embodying features of the present invention. The program instructions 114 comprise a specific technical

contribution to the art in accordance with the invention, as further described below.

[0029] With regard to the preceding overview of the DSP server 102, and other processing systems and devices described in this specification, terms such as 'processor', 'computer', and so forth, unless otherwise required by the context, should be understood as referring to a range of possible implementations of devices, apparatus and systems comprising a combination of hardware and software. This includes single-processor and multi-processor devices and apparatus, including portable devices, desktop computers, and various types of server systems, including cooperating hardware and software platforms that may be co-located or distributed. Physical processors may include general purpose CPUs, digital signal processors, graphics processing units (GPUs), and/or other hardware devices suitable for efficient execution of required programs and algorithms. Computing systems may include conventional personal computer architectures, or other general-purpose hardware platforms. Software may include open-source and/or commercially-available operating system software in combination with various application and service programs. Alternatively, computing or processing platforms may comprise custom hardware and/or software architectures. For enhanced scalability, computing and processing systems may comprise cloud computing platforms, enabling physical hardware resources to be allocated dynamically in response to service demands. While all of these variations fall within the scope of the present invention, for ease of explanation and understanding the exemplary embodiments described herein are based upon single-processor general-purpose computing platforms, commonly available operating system platforms, and/or widely available consumer products, such as desktop PCs, notebook or laptop PCs, smartphones, tablet computers, and so forth.

[0030] In particular, the term 'processing unit' is used in this specification (including the claims) to refer to any suitable combination of hardware and software configured to perform a particular defined task, such as accessing and

processing offline or online data, executing training steps of a machine learning model, or executing prediction steps of a machine learning model. Such a processing unit may comprise an executable code module executing at a single location on a single processing device, or may comprise cooperating executable code modules executing in multiple locations and/or on multiple processing devices. For example, in some embodiments of the invention classification and bid decision processing may be performed entirely by code executing on DSP server 102, while in other embodiments corresponding processing may be performed is a distributed manner over a plurality of DSP servers.

[0031] Software components, e.g. program instructions 114, embodying features of the invention may be developed using any suitable programming language, development environment, or combinations of languages and development environments, as will be familiar to persons skilled in the art of software engineering. For example, suitable software may be developed using the C programming language, the Java programming language, the C++ programming language, the Go programming language, and/or a range of languages suitable for implementation of network or web-based services, such as JavaScript, HTML, PHP, ASP, JSP, Ruby, Python, Perl, and so forth. These examples are not intended to be limiting, and it will be appreciated that convenient languages or development systems may be employed, in accordance with system requirements. The descriptions, block diagrams, flowcharts, and so forth, presented in this specification are provided, by way of example, to enable those skilled in the arts of software engineering and machine learning to understand and appreciate the features, nature, and scope of the invention, and to put one or more embodiments of the invention into effect by implementation of suitable software code in accordance with this disclosure without exercise of additional inventive ingenuity.

[0032] Returning to Figure 1, the system 100 further comprises additional DSP servers, e.g. 118, 120 that, in use, compete with DSP server 102 to bid for placement of advertising content within online ad slots offered via an ad

exchange server 122. The ad exchange server 122 implements a digital marketplace allowing advertisers and publishers of web sites and other online content to buy and sell advertising space in the form of a real-time, online auction in which each DSP server 102, 118, 120 is an automated, high-speed, bidder. The ad exchange server 122 comprises a database 124 in which it maintains details of online content providers (web servers) and advertisers (DSPs) for the purpose of operating a digital advertising marketplace. The functions of ad exchange platforms such as DoubleClick™ (owned by Google™), AppNexus™, Microsoft™ Ad Exchange™, and OpenX™, are well-known and will therefore not be described in further detail herein, otherwise than as is necessary to adequately illustrate the operation of embodiments of the invention.

[0033] The system 100 further includes user terminal devices, exemplified by terminal device 126. The terminal devices 126 may be, for example, desktop or portable PCs, smartphones, tablets, or other personal computing devices, and each comprise a processor 128 interfaced, e.g. via address/data bus 130, with volatile storage 132, non-volatile storage 134, and at least one data communications interface 136. The processor 128 is also interfaced to one or more user input/output (I/O) interfaces 140. The volatile storage 132 contains program instructions and transient data relating to the operation of the terminal device 126.

[0034] The terminal device storage 132, 134 may contain program and data content relevant to the normal operation of the device 126. This may include operating system programs and data (e.g. associated with a Windows, Android, iOS, MacOS, Linux, or other operating system), as well as other executable application software generally unrelated to the present invention. The storage 132 also includes program instructions 138 which, when executed by the processor 128 enable the terminal device to provide a user with access to online content. While many applications are known for providing such access, for simplicity in the present description it is assumed that the program instructions

138 implement a web browser having a graphical user interface (GUI) presented via the user I/O interface 140.

[0035] Accordingly, in the event that a user of the terminal device 126 access a web server 142, a corresponding web page display 144 is generated via the device UI 140. The display 144 include website content 146, and one or more advertising slots, e.g. 148, 150. As is further illustrated, each advertising slot 148, 150 may comprise a plurality of specific 'offers' on behalf of an advertiser. These offers are commonly arranged in a grid layout, e.g. as indicated by dashed rectangles 148a, 148b, 148c, 150a, 150b, 150c in Figure 1. A number of communications steps then take place in order to populate these slots, i.e. to provide online advertisers with ad impressions within the web page display 144. These communications steps will now be described with reference to the timeline 200 illustrated in Figure 2.

[0036] Initially, the user terminal 126, via the executing web browser application 138 and responsive to user input, transmits 202 an HTTP request to the web server 142 which includes a URL of desired web content. The web server 142 responds by transmitting 204 content, e.g. a web page in HTML format, to the user device 126. As will be appreciated by persons skilled in the art of web programming, the complete population and rendering of web page display 144 may require multiple requests and responses, and may involve further transactions with the web server 142 and/or with other online servers, such as content distribution network (CDN) servers and other web servers providing embedded content. For simplicity and to facilitate focus on communications embodying features of the present invention, all such known additional transactions are represented by a single exemplary communication 206 in Figure 2.

[0037] In order to obtain advertising content to fill the slots 148, 150, the web page transmitted by the web server 142 to the user device 126 typically includes a hypertext reference ('href') directing the browser 138 to retrieve content from

the ad exchange server 122 in accordance with an application programming interface (API) defined and provided by the relevant operator of the server 122. Accordingly, the user device 126 transmits 208 an HTTP request to the ad exchange server 122. The request includes web site information and user information relating to the user of the terminal device 126. Available user information may include information that the web server 142 has gathered, and may include client-side information, such as device and browser identity and technical details, identifying information and contents of browser cookies, and the like. Many online mechanisms for gathering, maintaining, and tracking user and device information are well-known and available to persons skilled in the art of web programming, and will therefore not be described in further detail here.

[0038] The ad exchange server 122 receives the request, identifies relevant DSP servers 102, 118, 120 in its database 124, and transmits 210 bid request messages to each selected DSP server. One such bid request message, including site and user information, is received at DSP server 102 embodying the present invention, which executes a process 212 in accordance with its specific programming 114 in order to predict a likelihood of user interaction with a selected ad including one or more offers, placed within one or more of the available slots 148, 150, and arrive at a bid decision. In the event that a decision is made to bid for the offered impression, and a bid value determined, the DSP server 102 then transmits 214 the bid to the ad exchange server 122.

[0039] The ad exchange server 122 receives all bids transmitted from DSP servers, including server 102, and selects a winning bid. It then retrieves ad content corresponding with the winning bid from its database 124, and transmits 216 the ad content to the user device 126 for rendering within the corresponding ad slot, e.g. 148 or 150.

[0040] It is well-known that page load speed is an important characteristic of a web site from the user's perspective, and that it is undesirable for the time required for a web page to load fully to be excessive. Typically, it is preferred that

load time not exceed a few seconds, e.g. 3 seconds 218. There are, as has been described above, many steps necessary to fully serve all content of a complex web page, which may involve multiple servers across the global internet. It is, accordingly, critical that duration of the bidding process facilitated by the ad exchange server 202 be strictly limited. It is presently considered desirable that the DSP server 102 should make a bid decision in no more than a few tens of milliseconds, for example in under 30 milliseconds 220. This decision must be made with limited user information, and in view of the fact that a bad decision may have significant consequences for the advertiser. For example, if the DSP server wrongly determines that the user is a desirable target for a particular ad (i.e. computes a 'false positive'), it may place a relatively high winning bid and incur a real cost with little or no prospect of any return. Conversely, if the DSP server wrongly determines that the user is not a desirable target for the ad (i.e. computes a 'false negative'), it may place no bid, or a low losing bid, and cause the advertiser to miss an opportunity to obtain an impression with a real prospect of a return.

[0041] In order to achieve quality decision-making at high speed in the context of travel booking services, embodiments of the present invention employ a machine learning approach. To further facilitate understanding of this approach, reference is now made back to Figure 1, in which the system 100 further includes a machine learning server ('ML server') 152, which is configured to process raw data relating to placement of content (i.e. ads/offers) along with user interactions (i.e. user clicks on ads/offers), to generate training data sets for a machine learning model, and to train the machine learning model for deployment to the DSP server 102. The processing, training and deployment steps are described in greater detail below, with reference to Figures 3 and 4, and may be carried out continuously, periodically and/or on-demand in order to maintain currency of the machine learning model.

[0042] As with the DSP server 102, the ML server 152 may comprise a computer system having a conventional architecture, e.g. comprising a processor

154 that is operably associated with a non-volatile memory/storage device 156, via one or more data/address busses 158 as shown. The processor 154 is also interfaced to volatile storage 160 which contains program instructions and transient data relating to the operation of the ML server 152. Conventionally, the storage device 156 contains operating system programs and data, as well as other executable application software necessary for the intended functions of the ML server 152, and including program instructions which, when executed by the processor 154, cause the ML server 152 to perform operations relating to an embodiment of the present invention, such as are described in greater detail below with reference to Figures 3 and 4 in particular. In operation, instructions and data held on the storage device 156 are transferred to volatile memory 150 for execution on demand. Additionally, the processor 154 is operably associated with a communications interface 162 in a conventional manner, providing access to the Internet 116.

[0043] In use, the volatile storage 160 contains a corresponding body 164 of program instructions transferred from the storage device 156 and configured to perform processing, training and deployment steps embodying the present invention. The program instructions 164 comprise a further specific technical contribution to the art in accordance with the invention.

[0044] The system 100 further includes at least one database 166, which is configured to store raw historical data relating to placement of content (i.e. ads/offers) along with user interactions (i.e. user clicks on ads/offers). The volume of such data may be very large over time periods of interest, such as one month or more. For example, in a particular live deployment, it was found that a log of data for a single day comprises on the order of 20 million lines (i.e. placement and interaction events) having a total storage size on the order of 10 Gb. Accordingly, the database 166 is preferably implemented using technologies that are optimised for efficient storage, retrieval and update of very large volumes of data (sometimes referred to as 'big data') across multiple database servers and storage devices. While a number of suitable commercial

and open source technologies exist for implementation of the database 166, an exemplary experimental embodiment has been implemented using Apache Hadoop framework, with data stored in Parquet format on HDFS (Hadoop Distributed File System), and using Impala to provide a high-speed, SQL-like query engine. This implementation has been tested and found to provide more than adequate performance for practical online deployment of embodiments of the invention.

[0045] The database 166 is accessible to both the DSP server 102 and the ML server 152. In Figure 1, logical access is illustrated by corresponding arrows. In a practical embodiment, physical access between the database 166 and the DSP and ML servers 102, 152 may be via the Internet 116, and/or via other dedicated communications links or networks, such as a local storage area network (SAN). The DSP server 102 is configured to update the database 166, in real time, with raw data relating to placement and interaction events. The ML server 152 is configured to retrieve the raw data from the database 166 and to carry out processing, training and deployment steps, based on the retrieved data, in accordance with an embodiment of the invention.

[0046] Returning to Figure 2, further operations relating to update of the database 166 by the DSP server 102 are illustrated. In particular, in the event that the DSP server 102 places a successful bid, and corresponding ad content is transmitted 216 to the user device 126, the DSP server 102 updates 222 the database 166, adding data relating to the placement of the ad (i.e. ad/offer impression). Code associated with the ad is configured such that, in the event that the user subsequently interacts with (i.e. clicks on) the ad, the DSP server 102 receives, either directly or indirectly, a notification 224 of this interaction event. The DSP server 102 then updates 226 the database 166 with details of the interaction event. In this way, the database 166 is continuously updated with raw data relating to all placement and interactions events known to the DSP server 102.

[0047] Figure 3 is a block diagram illustrating schematically a number of code modules that together comprise an online user interaction prediction engine 300 embodying the invention. Implementation of the user interaction prediction engine 300 is distributed across the ML server 152 and DSP server 102, as shown by the dashed boxes in Figure 3. Three code modules make up the ML server component of the engine 300, namely a matching module 302, a feature enrichment module 304 and a machine learning module 306. These three modules are all implemented within the program instructions 164 executing on the ML server 152. The functionality implemented within each of these modules will now be described in greater detail.

[0048] The purpose of the matching module 302 is to match placement events (i.e. display of ads, and offers within ads, in ad slots 148, 150 of the display 144 of the user device 126) to subsequent interaction events (i.e. instances of a user clicking on an offer within an ad placed on the display 144 of the user device 126). Matching enables placement events to be tagged as 'clicked' or 'not clicked', so that they can be used by machine learning module 306 in training of a supervised machine learning model for prediction of user interaction events based upon placement event data. Additionally, matching enables placement event data to be combined with corresponding interaction event data to create a record for clicked ads containing all available information regarding placement and interaction.

[0049] Matching presents a challenge because there is no explicit link between a placement event (ad impression) and a subsequent user interaction (ad click). As illustrated in the time line 200 of Figure 2, a user interaction may occur at any time following placement, e.g. following a substantial delay. Since new placement and/or interaction events may occur at a very high rate (e.g. hundreds or thousands of times per second) in a live system, corresponding placement and interaction events may become widely separated in the database 166. Additionally, the rate of interaction events may be very low, e.g. it is generally reported that the click through rate (CTR) for web-based display

advertising is on the order of 0.05%. Furthermore, it is desirable to link placement and interaction events at offer level, rather than only at ad level.

[0050] The general approach employed for matching in embodiments of the invention is to identify, in the database 166, placement events and subsequent interaction events within a predetermined time window that have a selected set of matching parameters. The time window should be of sufficient duration to capture a substantial majority of all interactions, and the number and choice of parameters should be sufficient to ensure unique matching in a substantial majority of cases. Perfect matching may be difficult to achieve, because it is impossible to know if or when an interaction will occur. A time window of longer duration will capture interactions that occur after longer delays, but will also increase the risk of erroneous matching where, for example, a user interacts with a subsequently-presented ad having similar parameters. Similarly, the risk of erroneous matching can be reduced by using a larger selected set of parameters to distinguish between presented ads, at the expense of making the matching process more complex.

[0051] In exemplary experimental embodiment, the invention has been implemented in the context of a domain-specific DSP server operating on behalf of advertisers, using event data captured from a live system. A heuristic approach was taken to design of the matching module, with a number of experiments being conducted to determine a suitable time window, and a selected set of parameters. An 80 second time window was found to be effective in combination with matching the following event parameters:

- unique user identifier (tracked via a browser cookie);
- advertiser identifier;
- publisher identifier (i.e. the ad exchange/distribution network through which the ad was placed);
- format of the clicked offer (e.g. width and height of offer graphic, in pixels);

- ad product type;
- ad product pool;
- user segment (a combination of a user product segment, based upon a product such as flight, hotel or restaurant previously viewed by the user, and a user time segment, indicating how long it has been since the last activity of the user);
- site URL;
- ad slot visibility;
- user device;
- a measure of distance between a destination (location) about which the user was seeking information and a destination that was the subject of a specific offer; and
- ad slot key (a stable identifier for the combination of publisher, ad slot and page).

[0052] In the exemplary embodiment, matching is performed using an Impala SQL query to select and join tables of records of placement and interaction events on the values of fields corresponding with the parameters listed above. Specifically, placement records are LEFT JOINed to interaction records, such that the resulting table includes a row for each placement event. Each row comprises a set of values of raw features derived from the matched events, along with an indicator of whether or not an interaction event, i.e. ad/offer click, occurred. The table of matched data is input to the feature enrichment module 304.

[0053] The function of the feature enrichment module 304 is to derive, from the values of raw features in the matched data table generated by the matching module 302, a corresponding set of enriched feature vectors for use by the machine learning module 306. A process for determining a suitable set of enriched features (i.e. feature engineering) is described in detail below with reference to Figure 5. In Figure 3, definitions of enriched features for use by the

feature enrichment module 304 are shown as being stored in a file 310 within data store 308, however this may be regarded as a schematic convenience. In a practical embodiment, feature definitions may be stored in this way, may be compiled into a code module and linked to the feature enrichment module 304, or may be hard-coded into the feature enrichment module. As will be appreciated, each of these implementation options (and others that will be apparent to persons skilled in the art) potentially offers a different trade-off between flexibility, code complexity and execution speed.

[0054] In the exemplary embodiment, all of the enriched features are of categorical type (i.e. take on one of a number of discrete values), and are one-hot encoded. The resulting feature vectors are therefore generally relatively sparse, and comprise binary elements. Furthermore, each feature vector corresponds with an offer within an ad presented to a user, and is associated with a binary tag indicating whether or not the user interacted with (i.e. clicked on) the offer. The resulting table of feature vectors and tags is input to the machine learning module 306.

[0055] The machine learning module 306 comprises program code executing on the ML server 152, and configured in the exemplary experimental embodiment to implement a generalised linear model. Specifically, the machine learning module 306 of the exemplary embodiment implements a regularised logistic regression algorithm, with 'follow-the-regularised-leader' (FTRL)-proximal learning. Advantageously, this machine learning algorithm is known to be effective in the case of highly unbalanced datasets (noting that only around 0.05% of samples in the table of feature vectors are tagged as 'clicked'). Further details of the algorithm, and its application to click-prediction, can be found in H. Brendan McMahan *et al*, 'Ad Click Prediction: a View from the Trenches', *KDD'13*, August 11–14, 2013, Chicago, Illinois, USA. The algorithm has a number of hyperparameters that can be adjusted in order to optimise its learning accuracy on the training data for a specific problem. A process for determining a suitable set of values for the hyperparameters is described in detail below with

reference to Figure 5. In Figure 3, fixed values of the hyperparameters for use by the machine learning module 306 are shown as being stored in a file 312 within data store 308. As will be appreciated, however, alternative implementations are possible, such as hard-coding the parameters into the machine learning module 306.

[0056] Execution of the machine learning module 306 on a particular dataset results in the generation of a model that can be executed by the DSP server 102, as will be described in greater detail below with reference to Figure 6. In particular, a logistic regression model is wholly characterised by a set of coefficients associated with elements of the input feature vector. In the exemplary embodiment, a particularly efficient representation of the model is employed, to enable the DSP server 102 to compute a prediction of the likelihood of a user interaction very rapidly, i.e. well within the 30 ms target window 220 for generating a bid decision. Specifically, the coefficients are stored in a dictionary data structure in which each entry is defined by a key and a value. The key is a hashed representation of a concatenation of the feature name (i.e. column label in the feature table) and a corresponding feature value (i.e. categorical values prior to one-hot coding). The associated value in the dictionary is simply the corresponding model coefficient. This type of data structure is known to provide extremely fast lookup, particularly for sparse feature sets. In particular, by using hashed values a limit on the number of hashed features may be imposed (a scheme sometimes referred to as the 'hashing trick'). This scheme can be used to greatly speed lookup and computation, at the expense of possible collisions in dictionary key values. Advantageously, however, the statistical effect of these collisions can be neglected from the perspective of overall performance of the algorithm.

[0057] For deployment to the DSP server 102, the model data structure is serialised in a binary format (in the exemplary embodiment the Python 'pickle' format is used), and stored in a model file 314 in data store 308.

[0058] In use, the ML server 152 executes the modules 302, 304, 306 repeatedly, e.g. continuously, periodically, or on-demand. This is illustrated by the flowchart 400 shown in Figure 4. Raw data is retrieved from the database 166 at step 402. Exemplary embodiments use a predetermined period of recent data, which is considered to be representative of the behaviour of current online users of the system 100. For example, raw data from the most recent one-month period may be employed. At step 404, the matching module 302 performs matching of placement and interaction events, as has been described. In practice, retrieval 402 and matching 404 steps may be combined as a single query, e.g. an Impala SQL query.

[0059] At step 406, the ML server 152 executes the feature enrichment module, which uses the enriched feature definitions 310 to compute enriched feature vectors corresponding with the matched data. These are transferred to the machine learning module 306 which trains the model using the tagged feature vectors and the predetermined hyperparameters defined in the configuration file 312. The resulting model coefficients are hashed, serialised and published 410 to the model file 314.

[0060] Optionally, the ML server then waits 412, before recommencing the process at step 402. Exit from the wait condition 412 may be triggered by a number of different events. For example, the ML server may be configured to run the modules 302, 304, 306 periodically, e.g. once per day. Alternatively, or additionally, it may be configured to run the modules 302, 304, 306 on-demand, e.g. upon receipt of a signal from a controller (not shown) within the system 100. In some embodiments the ML server may run the modules 302, 304, 306 continuously, thereby updating the model file 314 as frequently as possible based upon the time required for data matching, feature enrichment and model training. In an exemplary experimental configuration, it was found that updates based upon 30 minute batches of data provided a suitable trade-off between quality of the output of the matching module 302 (i.e. the need to reconcile interaction and placement events accurately for a good training dataset), and reactivity to the

real-time changes in the ad exchange network (e.g. new campaign launches, entry/exit of competitors, changes in user demand for some contents, and so forth).

[0061] Turning now to Figure 5 there is shown a flowchart 500 of a process of feature engineering and model hyperparameter optimisation according to an embodiment of the invention. In practice, the process 500 is partially automated, and operated under human supervision. The development of suitable features with strong predictive capability, and the selection of appropriate ranges of model hyperparameters involves significant experience, judgment, creativity and ingenuity, and in most cases cannot efficiently be fully-automated.

[0062] The process 500 requires a set of test data, which is retrieved at step 502, and which may be obtained in the same manner as described above in relation to the functionality of the matching module 302. In particular, data may be extracted from the database 166 for a selected test period using an Impala SQL query of the same form as that used by the matching module 302.

[0063] At step 504, a set of enriched features is defined and configured. In the exemplary embodiment, this step involves application of judgment, creativity and ingenuity of an experienced data scientist. In practice, a number of experiments have been performed, according to the process 500 and supported by further analysis of the test data set, in order to identify an effective set of enriched features. At step 506, values of the defined enriched features are computed from the raw test data set.

[0064] At step 508, a set of hyperparameter values is selected and a machine learning model is configured with the selected values. At step 510 the resulting model is trained using the enriched test data. Typically, a portion of the test data is held back in the training step 510, which is then used in a cross-validation step 512 to assess the performance of the trained model on data that was not seen during the training step 510.

[0065] Performance of the trained model is then assessed at decision step 514, to determine whether or not it is acceptable, for example by reaching some optimal or sufficient level of performance. The choice of criteria for assessing performance may be important to identifying an acceptable model. Various known criteria may be employed, such as Area Under the Receiver Operating Curve (AUROC), log loss, or Gini (which is related to the AUROC). In the exemplary embodiment, a combination of Gini (which takes values between -1 and 1, and is desirably as high as possible) and log loss (which is desirably as low as possible) was used to assess performance of different models. This approach was employed not only for different hyperparameters of the selected FTRL-Proximal model, but also for a number of alternative models, including decision trees (distributed random forest, gradient boosted trees), naïve Bayes, and deep learning networks, which were consequently rejected as providing inferior performance on the analysed datasets.

[0066] In the event that performance is deemed unacceptable, or an optimisation process is incomplete, at decision 514, a further decision 516 is made as to whether the model hyperparameters. The resulting loop of configuring hyperparameters, training and testing the model is typically automated using an algorithm such as grid search, or similar. The role of the supervising data scientist in this case is to determine suitable ranges for the grid of hyperparameters.

[0067] In the event that no further variation of hyperparameters is required, an outer loop, implemented via decision 518, allows for the testing of alternative sets of enriched features. If available selections and values of model algorithms, hyperparameters and enriched features have been exhausted without identifying an acceptable model, then the process 500 may be regarded as having failed, and a reconsideration of strategy may be required. For the purposes of the exemplary embodiment, however, the process 500 led to a model with acceptable performance. At step 520, therefore, the identified enriched feature definitions and model hyperparameters are written to the data files 310, 312 in the data store

308. A summary of the enriched features developed via the process 500 is presented in Table 1.

[0068] Returning to Figure 3, the online user interaction prediction engine 300 includes a real-time bidding module 316, which is implemented within the program instructions 114 executing on the DSP server 102. The real-time bidding module 316 employs the enriched feature definitions 310 and the trained model representation 314. In particular, the operation of the real-time bidding module 316 is represented by the flow chart 600 shown in Figure 6 in which, at step 602, site and user information is received, i.e. via transmission 210 from the ad exchange server 122. This information is used at step 604 to compute a corresponding enriched feature vector according to the definitions 310.

[0069] At step 606, the real-time bidding module accesses the model representation which, as has been described, comprises a set of coefficients stored in a highly efficient dictionary structure for rapid coefficient lookup. As described above, with reference to Figure 4 in particular, the model may be updated from time-to-time by the ML server 152. The model representation 314 may be stored in a shared storage medium 308, and be asynchronously readable by the DSP server 102. In some embodiments, the DSP server may maintain a cached copy of the model representation 314 for rapid access, which is updated upon update of the stored file by the ML server 152.

[0070] The output of the model is an estimate of likelihood of user interaction with an offer within a selected ad, based on the enriched feature vector. In the exemplary embodiment, the output is a value representing a probability that the user will click on an offer within the selected ad. This value is used in a bid decision process at step 608. The process 608 may include determining whether or not to bid at all, and/or a determination of a particular price to bid for the available ad slot. For example, a threshold may be applied, such that if the value is below the threshold then no bid is made. In some embodiments, a bid amount may be determined based upon the magnitude of the value, such that a higher

price is bid if the model indicates a higher likelihood that the user will click on an offer within the selected ad. In the event that a decision is made to bid for the slot, control is directed 610 to step 612 wherein the bid information is transmitted 214 back to the ad exchange server 122. In the event that the bid is successful, control is directed 614 to step 616, in which the database 166 is updated with details of the placement event.

[0071] In order to assess the performance of the real-time bidding module 316 embodying the invention, an experimental module was run in parallel with a number of modules implementing a conventional bidding algorithm. The results are shown in the charts of Figures 7(a) and 7(b).

[0072] In particular, Figure 7(a) is a chart 700 having click through rate (CTR) on the vertical axis 702, with the corresponding performance of ten bidding modules shown as a series of bars. The bars 704 represent the performance of nine conventional bidding modules, while the bar 706 represents the performance of the experimental bidder embodying the invention. As can be seen, the experimental bidder achieved a CTR of around five times the average performance of the conventional bidders.

[0073] Figure 7(b) is a chart 708 having margin, defined as net profit divided by cost, on the vertical axis 710. The bars 712 represent the performance of the nine conventional bidding modules, all of which operated at a loss. However, the experimental bidder, represented by the bar 714, was able to operate at a profit.

[0074] The results in Figures 7(a) and 7(b) thus clearly demonstrate the technical and practical superiority achievable by the invention over conventional methods of predicting online user interaction.

[0075] It should be appreciated that while particular embodiments and variations of the invention have been described herein, further modifications and alternatives will be apparent to persons skilled in the relevant arts. In particular,

the examples are offered by way of illustrating the principles of the invention, and to provide a number of specific methods and arrangements for putting those principles into effect. In general, embodiments of the invention rely upon providing technical arrangements whereby automated real-time online decision-making may be carried out based upon predictions of user interactions derived from a machine learning model trained using data derived from a database of placement and interaction events. Technical steps implemented by exemplary embodiments include matching of events to generate combined placement/interaction records that are tagged for use by supervised learning algorithms, calculation of enriched feature vectors for online learning, and training of a machine learning model based upon continuously updating event data to maintain a current and periodically-updating model representation having an efficient format usable by a real-time bidding module to make rapid decisions, e.g. in under 30 ms.

[0076] The described embodiments should be understood as being provided by way of example, for the purpose of teaching the general features and principles of the invention, but should not be understood as limiting the scope of the invention, which is as defined in the appended claims.

Table 1: Summary of enriched features

Feature Name	Feature Description
ts_day_of_week	The day of the week (Sun-Sat) of the placement event.
ts_hour_of_day	The hour of the day (00-23) of the placement event.
ts_is_weekend	Whether the placement event occurred on a weekend.
ts_is_bank_holiday	Whether the placement event occurred on a bank holiday in the country from which the user accessed a site.
publisher_id	Identifier of publisher (i.e. operator of ad exchange server).
advertiser_id	Identifier of advertiser.
offer_key	A unique offer identifier, created by combining advertiser_id (see above) and other advertiser fields (product type and product pool).
ad_dst_top199	A destination associated with an offer. Limited to the top 199 destinations, which were found in feature engineering experiments to capture 92% of all clicks.
fmt	Format of an offer (width and height of offer image within ad slot)
nb_offers_per_ad	Number of offers included with the ad slot.
mq_dst	Proximity/distance of destination of interest to the user and destination associated with an offer. A categorical value indicating closeness of match on a set scale.
user_pseg	Identifier of a product segment previously viewed by user (e.g. flight, accommodation, restaurant).
user_tseg	Identifier of a time segment of the user's previous activity (e.g. within last day, 24-48 hours ago, ..., 8-30 days ago).
domain_name_top99	Domain name of site in which ad slot is displayed. Limited to the top 99 domains, which were found in feature engineering experiments to capture 95% of all clicks.
slot_visibility	Visibility of ad slot within page on user display.
device	User device identifier.
fmt_device	An engineered feature comprising a combination of offer format (fmt) and user device identifier (device).
ad_slot_key_top499	A unique identifier for the combination of publisher, ad slot, and page. Limited to the top 499 values, which were found in feature engineering experiments to capture 97% of all clicks.
camp_type	Categorical identifier of campaign type associated with an offer (e.g. text + image, image, display banner with dynamic content, static display banner).

Feature Name	Feature Description
user_country_top3	The country from which the user accessed a site. Limited to the top three countries, which were found in feature engineering experiments to capture over 99% of all traffic. Note, however, that the number and identity of top countries is specific to a publisher/ad exchange, which may be region and language specific.
offer_pos	A categorical value indicating the placement of an offer within an ad slot.
browser	Identifier of user browser (e.g. Chrome, IE, Safari, etc).

CLAIMS:

1. A computing apparatus which implements a demand side platform, the computing apparatus comprising:

a processor;

at least one memory device accessible by the processor; and

a data communications interface operably associated with the processor,

wherein the memory device contains a body of program instructions including a machine learning model which is executable by the processor and configured to determine an estimate of likelihood of user interaction with a content item, the model having been trained using a set of enriched training feature vectors and corresponding interaction event tags derived from a matched data set generated from records relating to content placement events and records relating to user interaction events retrieved from an online data store wherein the placement and interaction events occur within a defined time period,

the body of program instructions further including instructions which, when executed by the processor, cause the computing apparatus to implement a method comprising steps of:

receiving, via the data communications interface, information relating to an online content placement slot and information relating to a user to whom content within the online content placement slot will be displayed;

computing an enriched estimation feature vector based upon a content item selected for placement within the online content placement slot, the information relating to the user, and the information relating to the corresponding online content placement slot; and

executing the machine learning model to determine an estimate of likelihood of the user interacting with the selected content item, based upon the enriched estimation feature vector.

2. The apparatus of claim 1 wherein the machine learning model is a generalised linear model comprising a plurality of model coefficients.

3. The apparatus of claim 2 wherein the machine learning model is a logistic regression model.

4. The apparatus of claim 2 or 3 wherein the plurality of model coefficients is stored in a dictionary data structure in which each entry is defined by a key and a coefficient value, wherein each key comprises a hashed representation of a concatenation of a feature name and a corresponding feature value, and wherein the program instructions cause the computing apparatus to implement the step of executing the machine learning model by:

generating, for each feature value of the enriched estimation feature vector, a corresponding key;

retrieving from the dictionary data structure, for each generated key, a corresponding coefficient value; and

computing, using the enriched estimation feature vector and the retrieved coefficient values, the estimate of likelihood of the user interacting with the selected content item.

5. The apparatus of any one of claims 1 to 4 wherein:

the online content placement slot is an ad slot;

the information relating to the ad slot and information relating to the user to whom content within the ad slot will be displayed is received along with a bid request message transmitted from an ad exchange server;

the content item comprises at least one offer for placement within the ad slot, and

the body of program instructions further include instructions which, when executed by the processor, cause the computing apparatus to implement the method comprising further steps of:

transmitting, to the ad exchange server, a bid response message in reply to the bid request message;

responsive to receiving, from the ad exchange server, a successful bid notification, updating an online data store with content placement event data relating to placement of the content item; and

responsive to receiving a notification of a user interaction with the content item, updating the online data store with user interaction event data relating to the user interaction with the content item.

6. The apparatus of any one of claims 1 to 5, wherein the records relating to content placement events and records relating to user interaction events comprise no explicit link between each other.

7. A computing apparatus which implements training of a machine learning model configured to estimate of likelihood of user interaction with content items, the computing apparatus comprising:

a processor;

at least one memory device accessible by the processor; and

a data store accessible by the processor,

wherein the memory device contains a body of program instructions including instructions which, when executed by the processor, cause the computing apparatus to implement a method comprising steps of:

accessing the data store to retrieve records relating to content placement events, and records relating to user interaction events, wherein the placement and interaction events occur within a defined time period;

matching retrieved content placement event records with retrieved interaction event records to generate a matched data set which comprises a plurality of records, each record of the matched data set including a set of raw feature values derived from a content placement event along with an interaction event tag indicating whether or not an interaction event occurred corresponding with the content placement event;

computing, from the raw feature values, a corresponding set of

enriched training feature vectors; and

training the machine learning model using the enriched training feature vectors and corresponding interaction event tags.

8. The apparatus of claim 7 wherein the machine learning model is a generalised linear model comprising a plurality of model coefficients.

9. The apparatus of claim 8 wherein the machine learning model is a logistic regression model, and the program instructions cause the computing apparatus to implement the step of training the machine learning model using regularised logistic regression with 'follow-the-regularised-leader' (FTRL)-proximal learning.

10. The apparatus of claim 9 wherein the body of program instructions further include instructions which, when executed by the processor, cause the computing apparatus to implement the method comprising further steps, for each coefficient of the plurality of model coefficients, of:

generating a key comprising a hashed representation of a feature name and a feature value corresponding with the coefficient; and

storing, in a dictionary data structure, a value of the coefficient in association with the key,

whereby the dictionary data structure comprises an efficient encoding of the machine learning model.

11. The apparatus of any one of claims 7 to 10 wherein the body of program instructions includes instructions which, when executed by the processor, cause the computing apparatus to repeatedly execute the steps of accessing the online data store, matching the retrieved content placement event records with the retrieved interaction event records, computing enriched training feature vectors, and training the machine learning model, to update the machine learning model.

12. The apparatus of any one of claims 7 to 11, wherein the records relating to content placement events and records relating to user interaction events comprise no explicit link between each other.

13. A computer-implemented method comprising:

accessing an online data store to retrieve records relating to content placement events, and records relating to user interaction events, wherein the placement and interaction events occur within a defined time period;

matching retrieved content placement event records with retrieved interaction event records to generate a matched data set which comprises a plurality of records, each record of the matched data set including a set of raw feature values derived from a content placement event along with an interaction event tag indicating whether or not an interaction event occurred corresponding with the content placement event;

computing, from the raw feature values, a corresponding set of enriched training feature vectors;

training a machine learning model using the enriched training feature vectors and corresponding interaction event tags;

receiving, at a processor configured to execute the machine learning model, information relating to an online content placement slot and information relating to a user to whom content within the online content placement slot will be displayed;

computing, by the processor, an enriched estimation feature vector based upon a content item selected for placement within the online content placement slot, the information relating to the user, and the information relating to the online content placement slot;

determining, by the processor executing the machine learning model, an estimate of likelihood of the user interacting with the selected content item, based upon the enriched estimation feature vector.

14. The method of claim 13 wherein:

the online content placement slot is an ad slot;

the information relating to the ad slot and information relating to the user to whom content within the ad slot will be displayed is received along with a bid request message transmitted from an ad exchange server; and

the content item comprises at least one offer for placement within the ad slot.

15. The method of claim 14 further comprising:

transmitting, to the ad exchange server by the processor, a bid response message in reply to the bid request message;

receiving, by the processor from the ad exchange server, a successful bid notification; and

updating, by the processor, the online data store with content placement event data relating to placement of the content item.

16. The method of claim 15 further comprising:

receiving, by the processor, a notification of a user interaction with the content item; and

updating the online data store with user interaction event data relating to the user interaction with the content item.

17. The method of claim 16 wherein the steps of accessing the online data store, matching the retrieved content placement event records with the retrieved interaction event records, computing enriched training feature vectors, and training the machine learning model, are repeatedly executed to update the machine learning model.

18. The method of any one of claims 13 to 17, wherein the records relating to content placement events and records relating to user interaction events comprise no explicit link between each other.

19. A computer program comprising program code instructions for executing the steps of the method according to claims 13 to 18 when said program is executed on a computer.

5

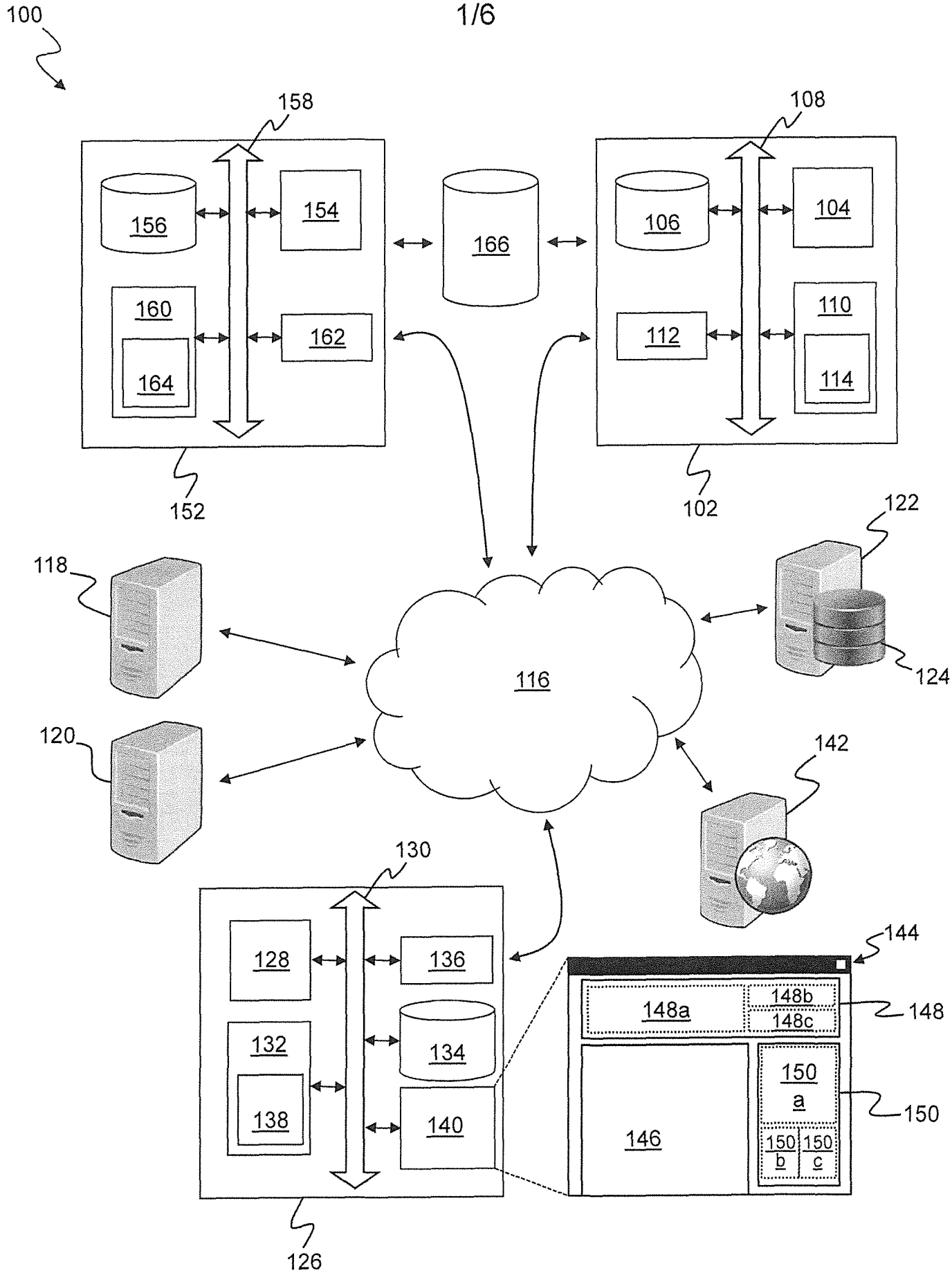
100
158
108
156
154
166
106
104
160
162
110
112
164
114
152
102
122
118
116
124
120
142
130
128
136
148a
148b
148c
148
132
134
150
a
138
140
146
150
150
b
150
c
144
126

Figure 1

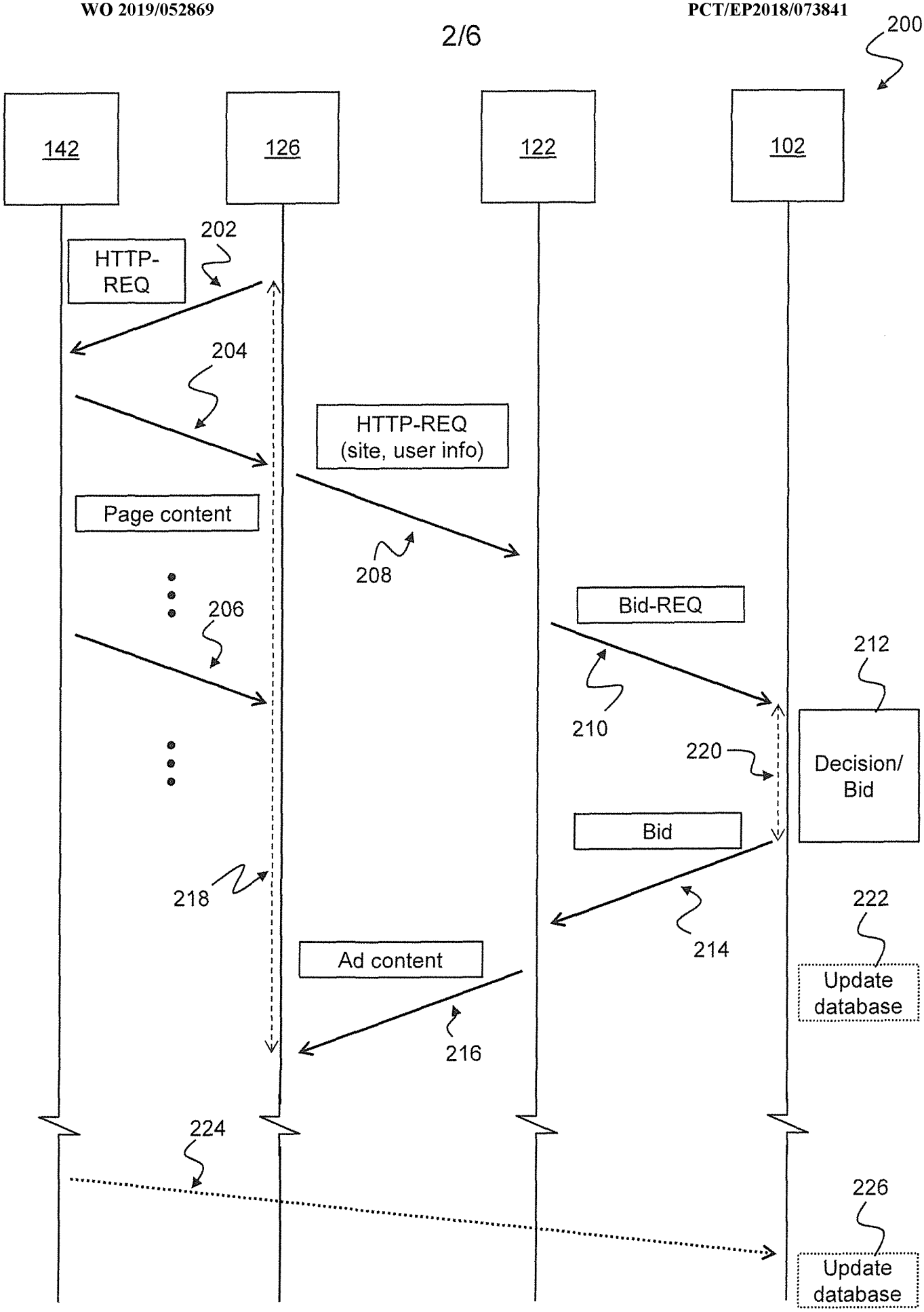
200
142
126
122
102
202
HTTP-REQ
204
HTTP-REQ (site, user info)
Page content
208
206
Bid-REQ
212
210
220
Decision/ Bid
Bid
218
214
222
Ad content
Update database
216
224
226
Update database

Figure 2

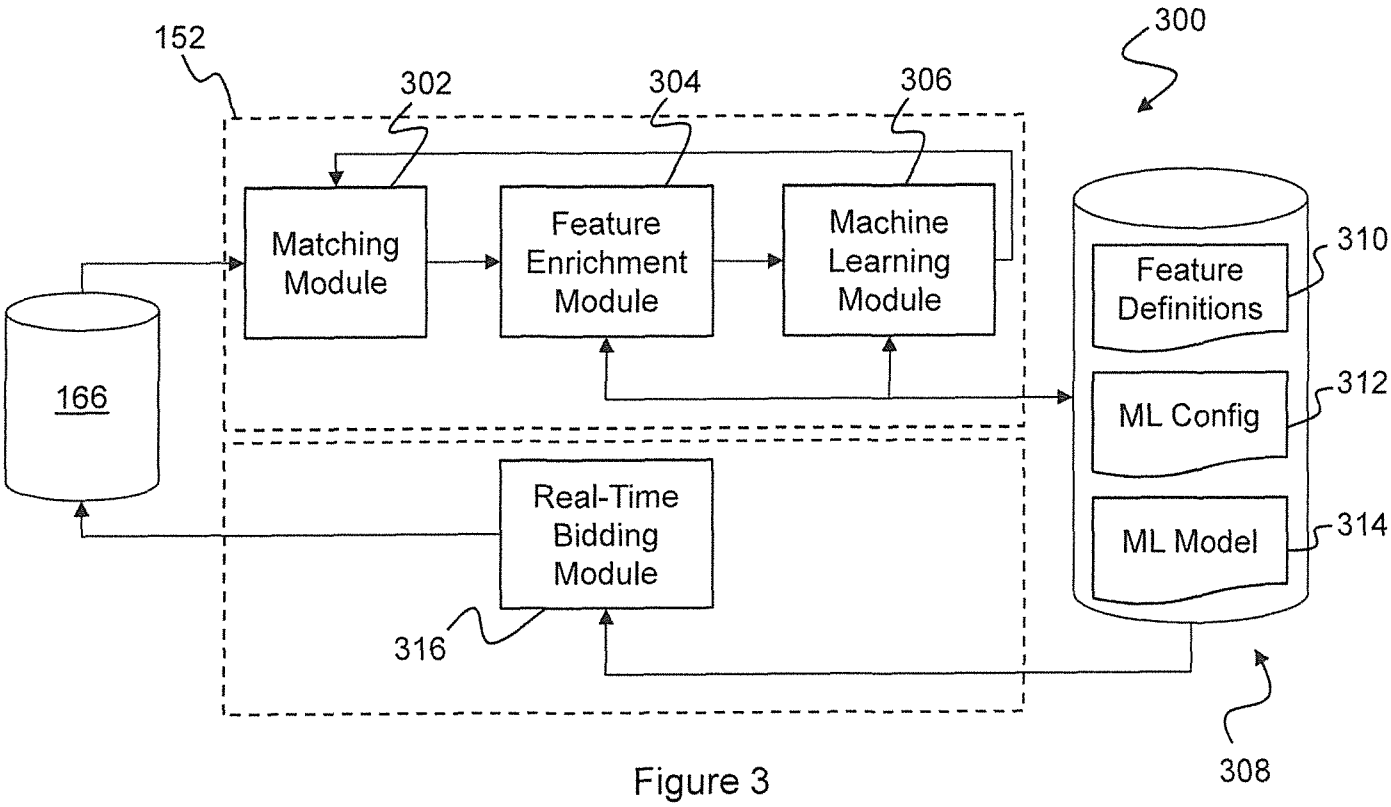

Figure 3

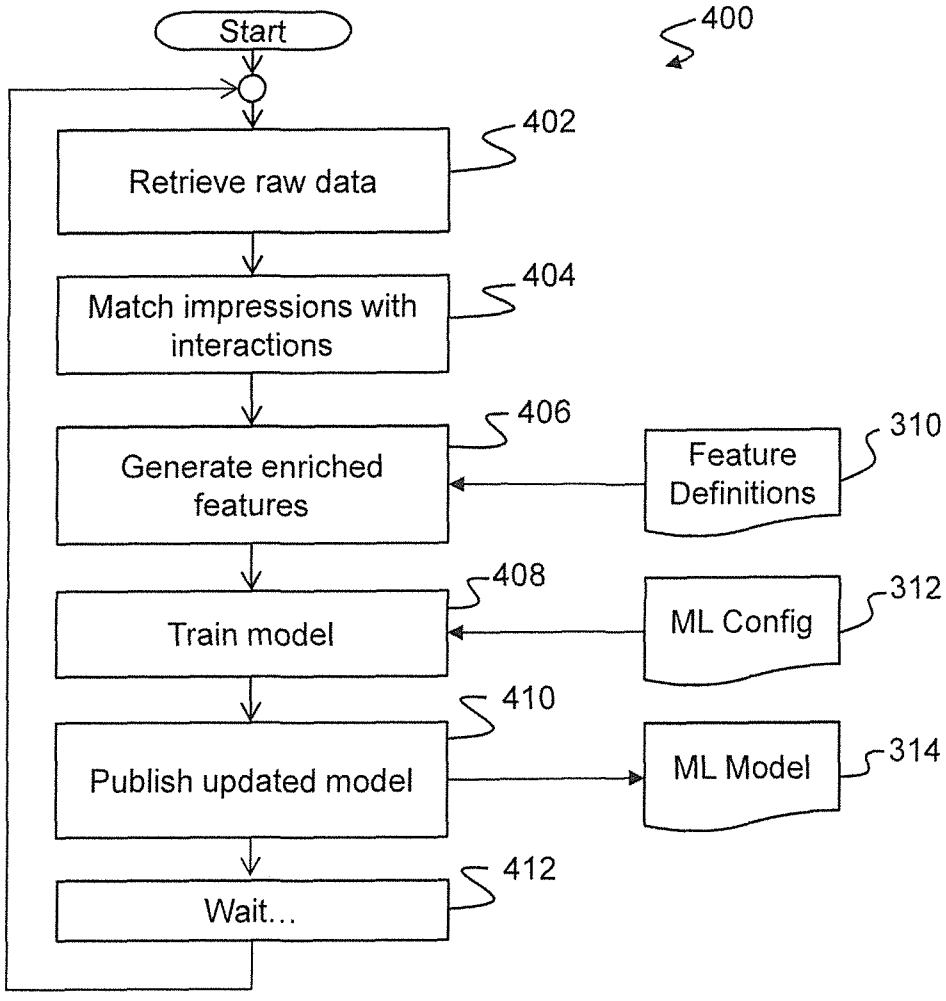

Figure 4

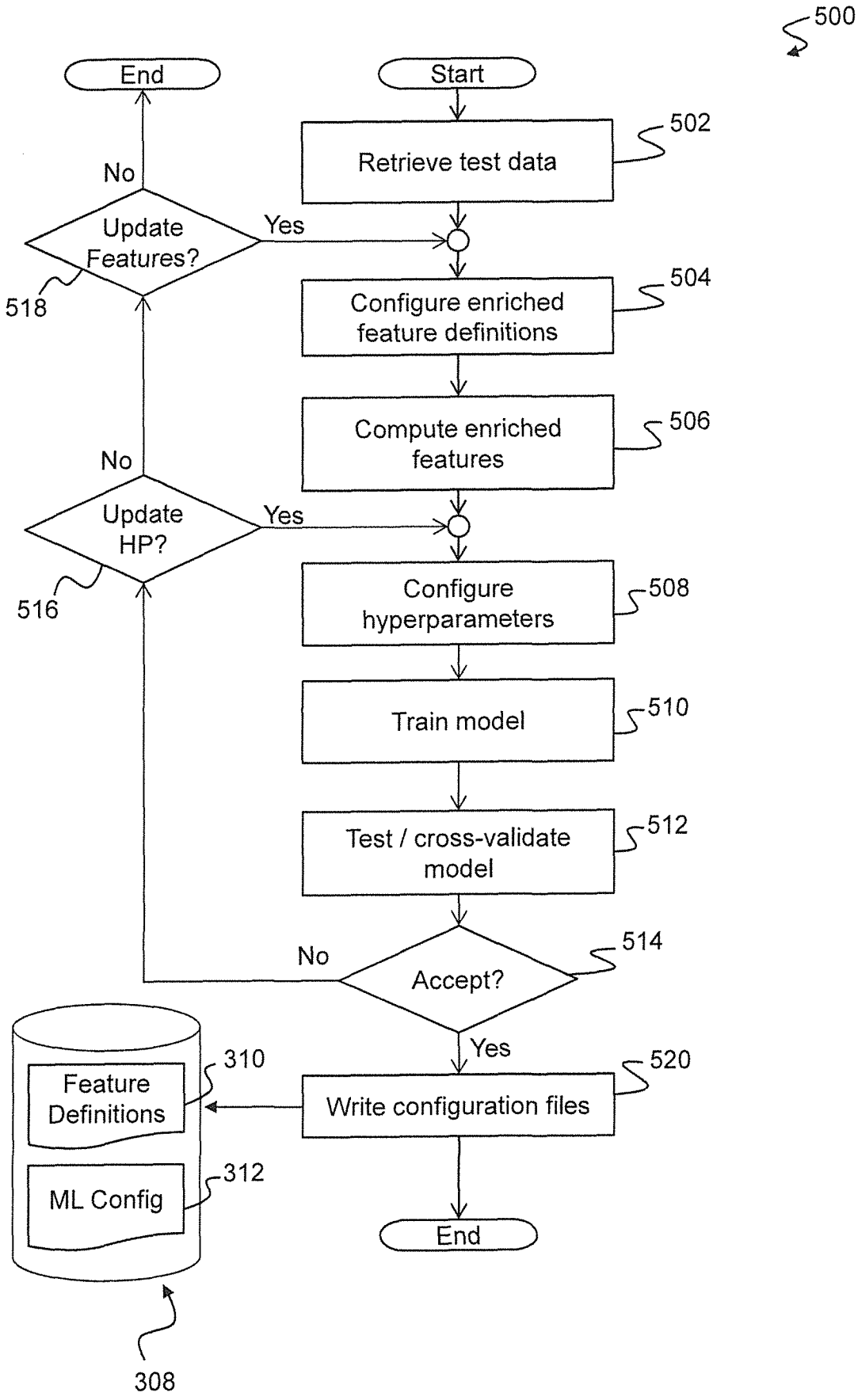
500
End
Start
Retrieve test data
502
No
Update Features?
Yes
518
Configure enriched feature definitions
504
Compute enriched features
506
No
Update HP?
Yes
516
Configure hyperparameters
508
Train model
510
Test / cross-validate model
512
No
Accept?
514
Yes
Feature Definitions
310
Write configuration files
520
ML Config
312
End
308

Figure 5

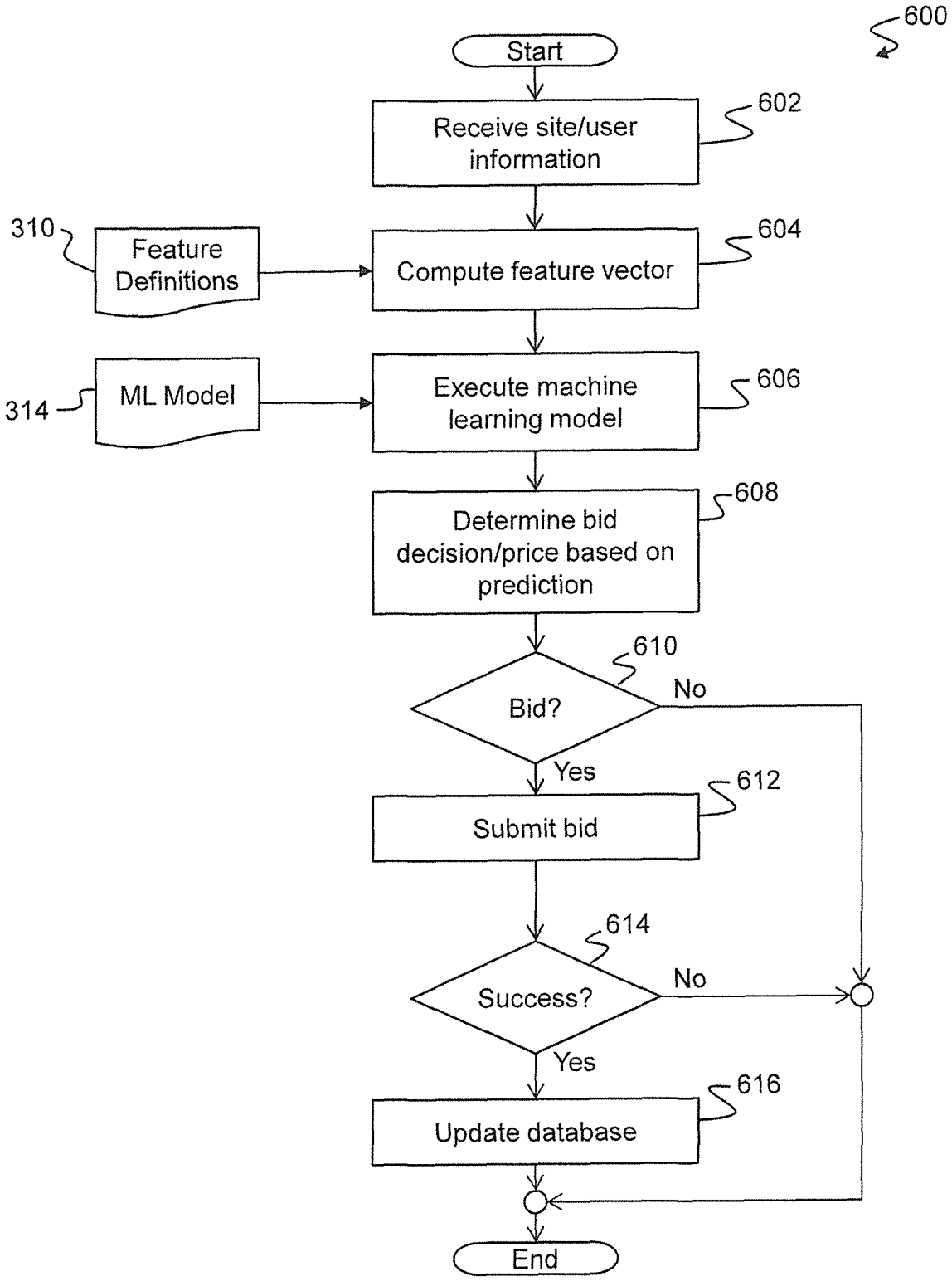
600
Start
Receive site/user information
602
310
Feature Definitions
Compute feature vector
604
314
ML Model
Execute machine learning model
606
Determine bid decision/price based on prediction
608
610
Bid?
No
Yes
612
Submit bid
614
Success?
No
Yes
616
Update database
End

Figure 6

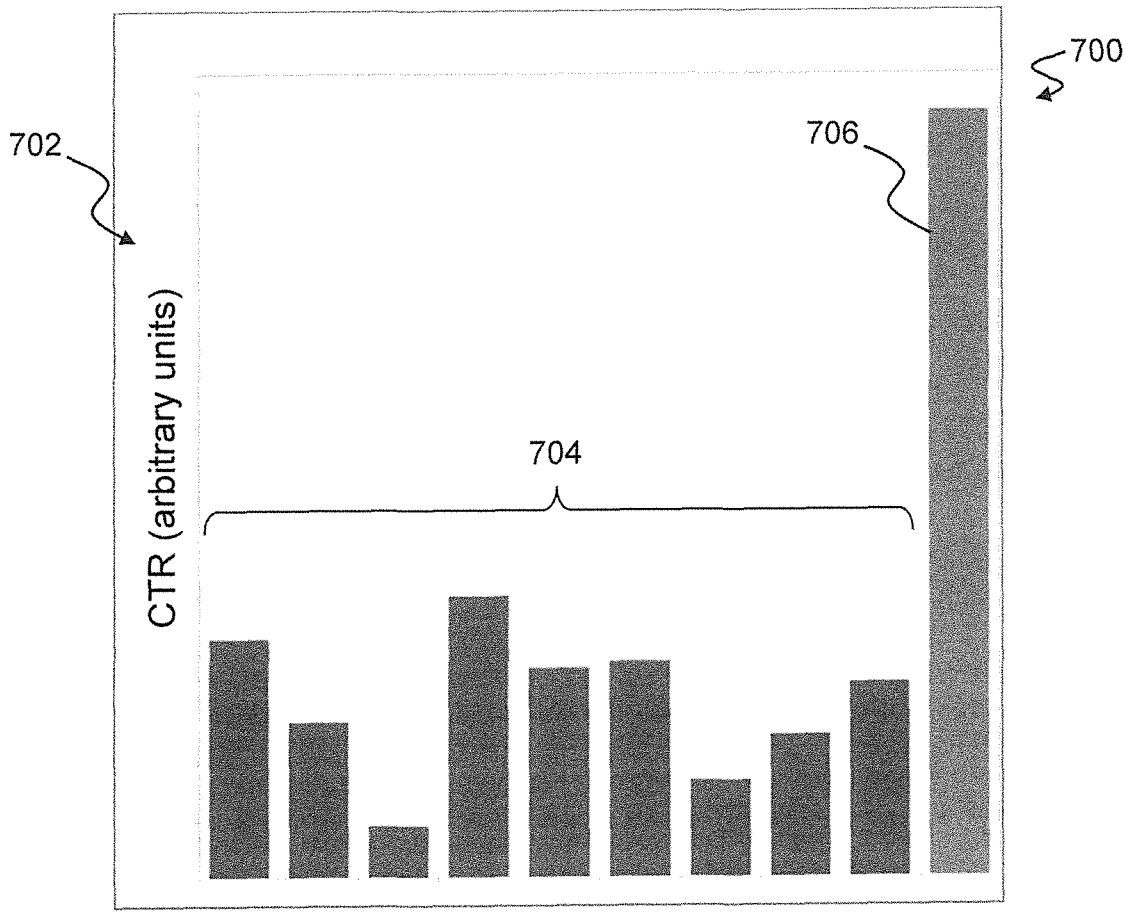

Figure 7(a)

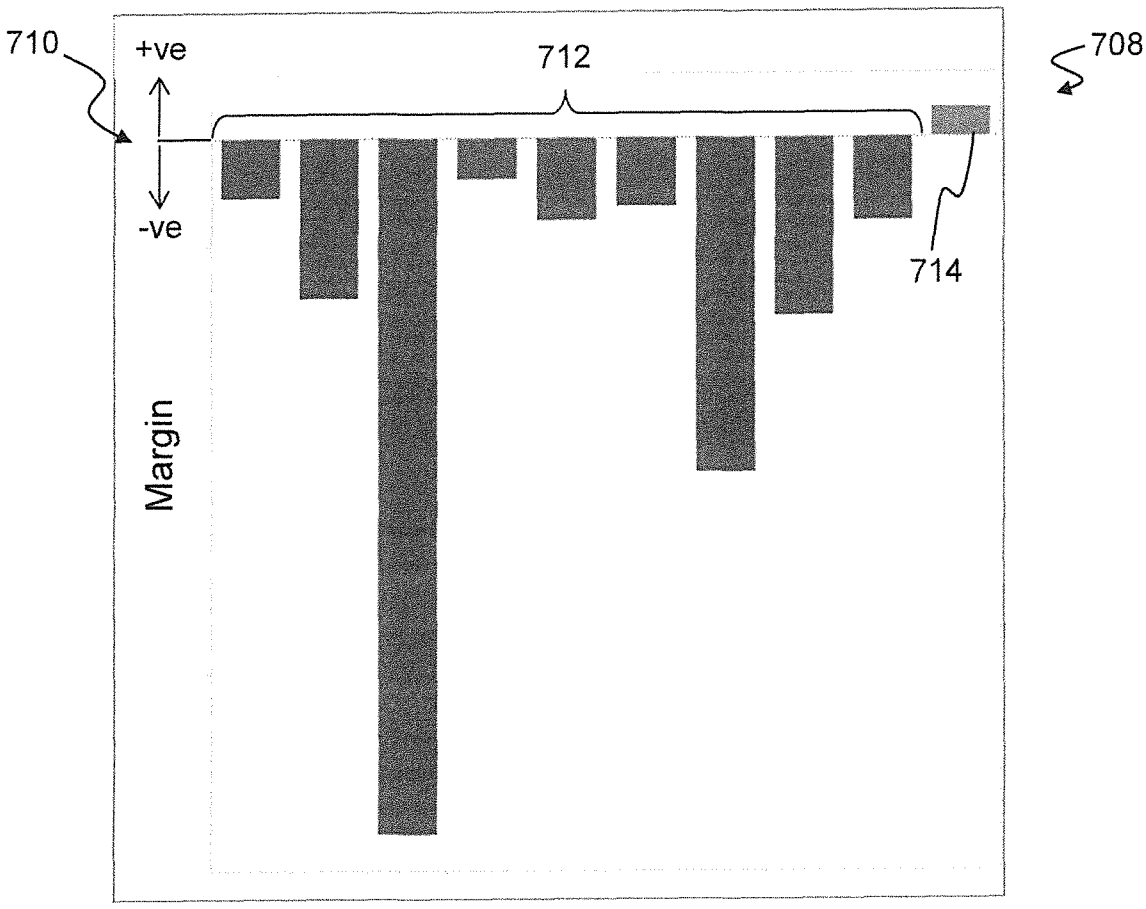

Figure 7(b)

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2018/073841

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06Q30/02 G06N5/00
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06Q G06N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2005/131762 A1 (BHARAT KRISHNA [US] ET AL) 16 June 2005 (2005-06-16) the whole document	1-19
X	US 2013/254787 A1 (COX EARL [US] ET AL) 26 September 2013 (2013-09-26) the whole document	1-19
X	EP 1 223 757 A2 (METABYTE NETWORKS INC [US]) 17 July 2002 (2002-07-17) the whole document	1-19
X	CN 101 484 890 A (IN V IDI TECHNOLOGIES CORP [US]) 15 July 2009 (2009-07-15) the whole document	1-19
X	JP 2016 062411 A (YAHOO JAPAN CORP) 25 April 2016 (2016-04-25) the whole document	1-19

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
10 October 2018	22/10/2018

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Guenov, Mihail

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2018/073841

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2005131762 A1	16-06-2005	AU 2004311786 A1	21-07-2005
		AU 2009222546 A1	22-10-2009
		BR PI0418343 A	02-05-2007
		CA 2552181 A1	21-07-2005
		CN 1922626 A	28-02-2007
		EP 1709577 A2	11-10-2006
		JP 5318349 B2	16-10-2013
		JP 2007517321 A	28-06-2007
		JP 2010176675 A	12-08-2010
		KR 20060130156 A	18-12-2006
		KR 20080103605 A	27-11-2008
		US 2005131762 A1	16-06-2005
		WO 2005065229 A2	21-07-2005
US 2013254787 A1	26-09-2013	US 2013254787 A1	26-09-2013
		US 2016142754 A1	19-05-2016
EP 1223757 A2	17-07-2002	AT 321422 T	15-04-2006
		DE 60209947 T2	22-02-2007
		EP 1223757 A2	17-07-2002
		ES 2261527 T3	16-11-2006
		JP 5283110 B2	04-09-2013
		JP 5421469 B2	19-02-2014
		JP 2003018584 A	17-01-2003
		JP 2008236794 A	02-10-2008
		JP 2013078150 A	25-04-2013
		US 2003101449 A1	29-05-2003
		US 2003101451 A1	29-05-2003
		US 2012011530 A1	12-01-2012
		US 2015143414 A1	21-05-2015
CN 101484890 A	15-07-2009	AU 2007247996 A1	15-11-2007
		CA 2651169 A1	15-11-2007
		CN 101484890 A	15-07-2009
		CN 107103489 A	29-08-2017
		EP 2024862 A2	18-02-2009
		JP 2009536413 A	08-10-2009
		US 2008059390 A1	06-03-2008
		US 2011067046 A1	17-03-2011
		US 2015264438 A1	17-09-2015
		WO 2007131069 A2	15-11-2007
JP 2016062411 A	25-04-2016	JP 6226846 B2	08-11-2017
		JP 2016062411 A	25-04-2016