



US 20170169345A1

(19) **United States**(12) **Patent Application Publication**
de Knijf et al.(10) **Pub. No.: US 2017/0169345 A1**(43) **Pub. Date: Jun. 15, 2017**(54) **PREDICTING CHURN FOR (MOBILE) APP USAGE**(71) Applicant: **AVG Netherlands B.V.**, Amsterdam (NL)(72) Inventors: **Jeroen de Knijf**, Amsterdam (NL);
Manuel de Francisco Vera, Haarlem (NL)(21) Appl. No.: **15/376,105**(22) Filed: **Dec. 12, 2016****Related U.S. Application Data**

(60) Provisional application No. 62/265,552, filed on Dec. 10, 2015.

Publication Classification(51) **Int. Cl.**
G06N 5/04 (2006.01)
G06N 7/00 (2006.01)
G06N 99/00 (2006.01)(52) **U.S. Cl.**
CPC **G06N 5/047** (2013.01); **G06N 99/005** (2013.01); **G06N 7/005** (2013.01)(57) **ABSTRACT**

A churn prediction model is presented that uses both behavioral data as well as user characteristics to predict whether a given user will churn (i.e., stop using) an application. Initially a training set of user interactions can be correlated to a churn probability value for various sequences of user activity. Then, as regards a real time user, user actions in navigating through the app may be recorded, and this information can be used, in addition to user characteristics, to predict the probability that this user will churn, thus implementing in a “nip churn in the bud” approach (or, the inverse, remain loyal and continue to use the app). In some embodiments, a partial set of user actions can be identified as subsequences of known churn sequences. To users performing those subsequences of activity, a real time message, offer or promotion may be sent so as to influence them not to churn. In exemplary embodiments of the present invention, user data may be uploaded from a user’s device to proprietary or cloud servers. Churn analysis, or a more detailed churn analysis, using up to the minute collective data for the given app, may, for example, be performed on those servers.

DISCRIMINATING VALUE	SCREEN VISITED	SCREEN VISITED	SCREEN VISITED	SCREEN VISITED
0.622503	OnboardVPN yes 0	Home Button: AddWifiNetwork		
0.600287	OnboardVPN yes 0	Home Wifi Off	Home Button SideMenu	
0.809708	Home Button: AddWifiNetwork	Home Button: AddWifiNetwork	Home Button SideMenu	
0.744413	OnboardVPN yes 0	Home Button AddWifiNetwork	Home Button AddWifiNetwork	
0.70858	OnboardVPN yes 0	Home Button: AddWifiNetwork	Home Button SideMenu	
0.615667	Bubble HotspotAutomation Dismissed	Bubble VPN Dismissed	SecureHotspot ContinueNoVPN 0	SecureHotspot yes 0
0.666932	Bubble HotspotAutomation Dismissed	Bubble VPN Dismissed	Home Button SideMenu	Home Hotspot wifiSettings
0.90919	Bubble HotspotAutomation Dismissed	Bubble VPN Dismissed	Home Button AddWifiNetwork	SecureHotspot ContinueNoVP N 0
0.714529	Bubble HotspotAutomation Dismissed	Bubble VPN Dismissed	Home Button AddWifiNetwork	SecureHotspot yes 0

FIG. 1

DISCRIMINATING VALUE	SCREEN VISITED	SCREEN VISITED	SCREEN VISITED	SCREEN VISITED
0.622503	OnboardVPN yes 0	Home Button AddWifiNetwork		
0.600287	OnboardVPN yes 0	Home Wifi Off	Home Button SideMenu	
0.809708	Home Button AddWifiNetwork	Home Button AddWifiNetwork	Home Button SideMenu	
0.744413	OnboardVPN yes 0	Home Button AddWifiNetwork	Home Button AddWifiNetwork	
0.70858	OnboardVPN yes 0	Home Button AddWifiNetwork	Home Button SideMenu	
0.615667	Bubble HotspotAutomation Dismissed	Bubble VPN Dismissed	SecureHotspot ContinueNoVPN 0	SecureHotspot yes 0
0.666932	Bubble HotspotAutomation Dismissed	Bubble VPN Dismissed	Home Button SideMenu	Home Hotspot wifiSettings
0.90919	Bubble HotspotAutomation Dismissed	Bubble VPN Dismissed	Home Button AddWifiNetwork	SecureHotspot ContinueNoVP N 0
0.714529	Bubble HotspotAutomation Dismissed	Bubble VPN Dismissed	Home Button AddWifiNetwork	SecureHotspot yes 0

FIG. 2

DISCRIMINATING VALUE	SCREEN VISITED	SCREEN VISITED	SCREEN VISITED	SCREEN VISITED
0.651903	OnboardWifi Start yes	Bubble HotspotAutomation Display		
0.856997	OnboardWifi Start yes	Home Wifi Off		
0.615102	Home Button SideMenu	SecureHotspot ContinueNoVPN 0		
0.649728	OnboardWifi Start yes	Bubble HotspotAutomation Display	Bubble VPN Display	
0.636087	OnboardWifi Start yes	OnboardVPN yes 0	Home Button AddWifiNetwork	Bubble AssistantOff Display
0.666401	OnboardWifi Start yes	OnboardVPN yes 0	Home Hotspot Connect	Bubble HotspotAutomation Display
0.727036	OnboardWifi Start yes	OnboardVPN yes 0	SecureHotspot ContinueNoVPN 0	SecureHotspot ContinueNoVPN 0
0.649728	OnboardWifi Start yes	Bubble HotspotAutomation Display	Bubble VPN Display	Bubble HotspotAutomation Display

FIG. 3

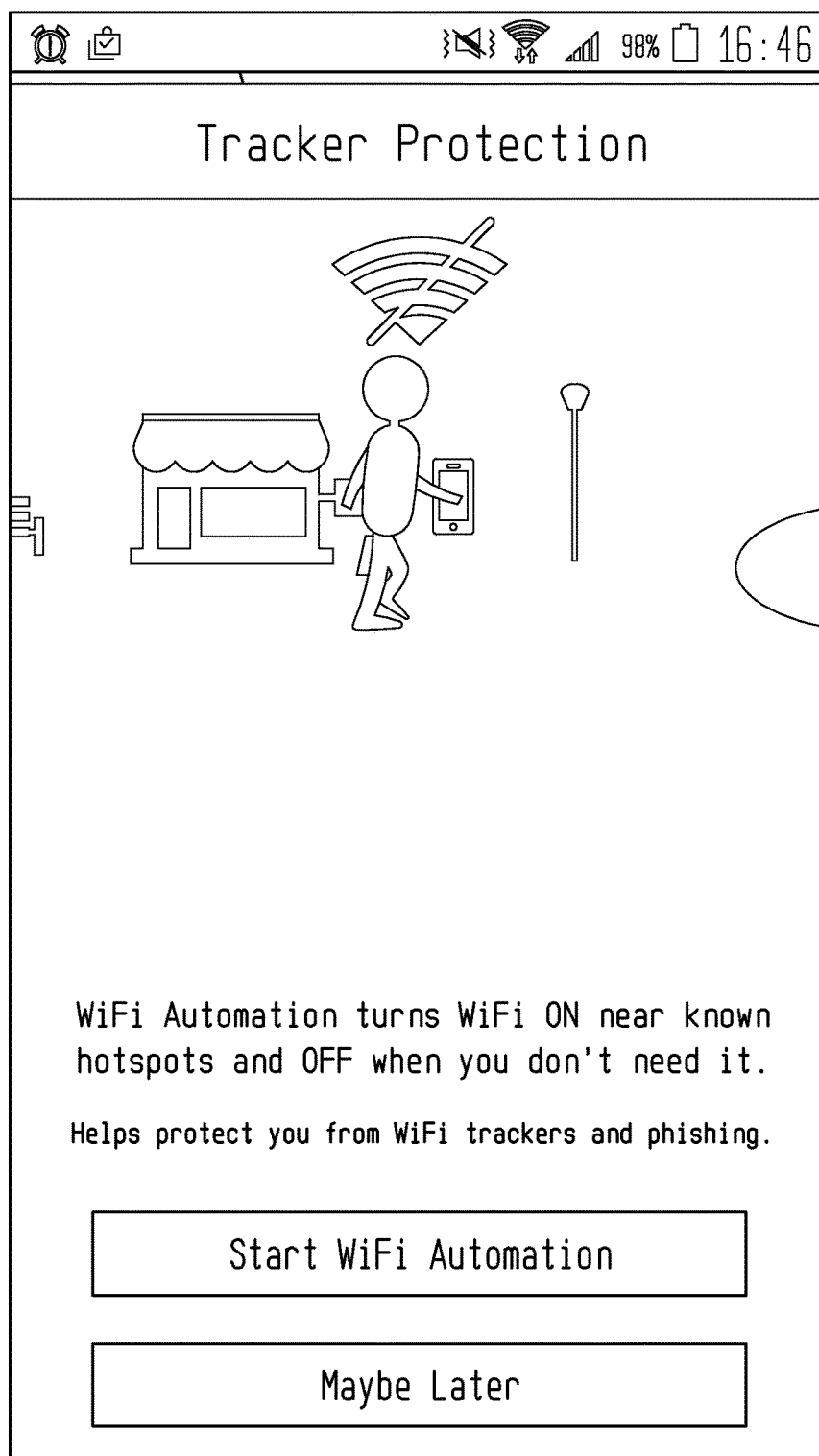
FIG. 4

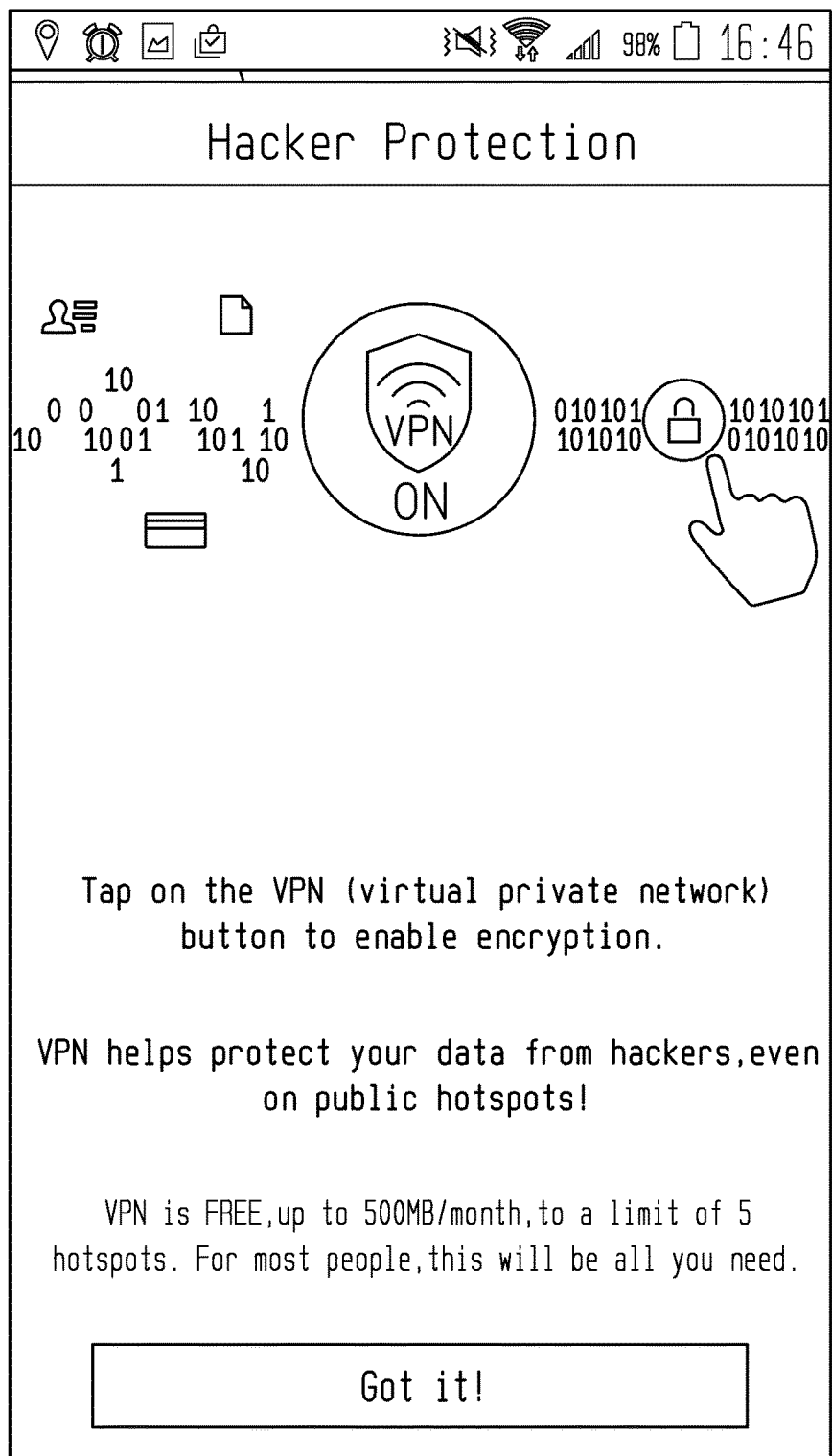
FIG. 5

FIG. 6

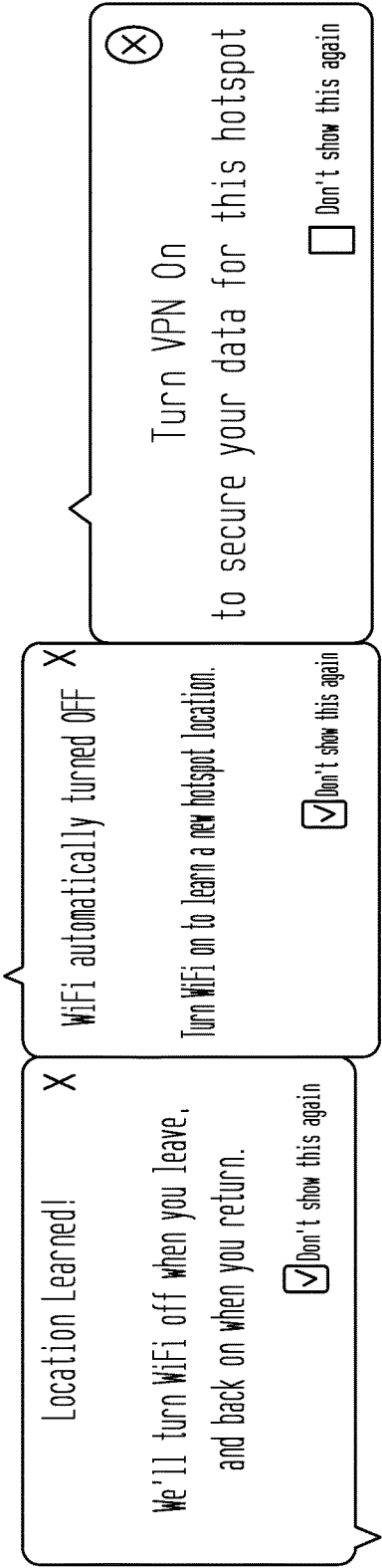


FIG. 7

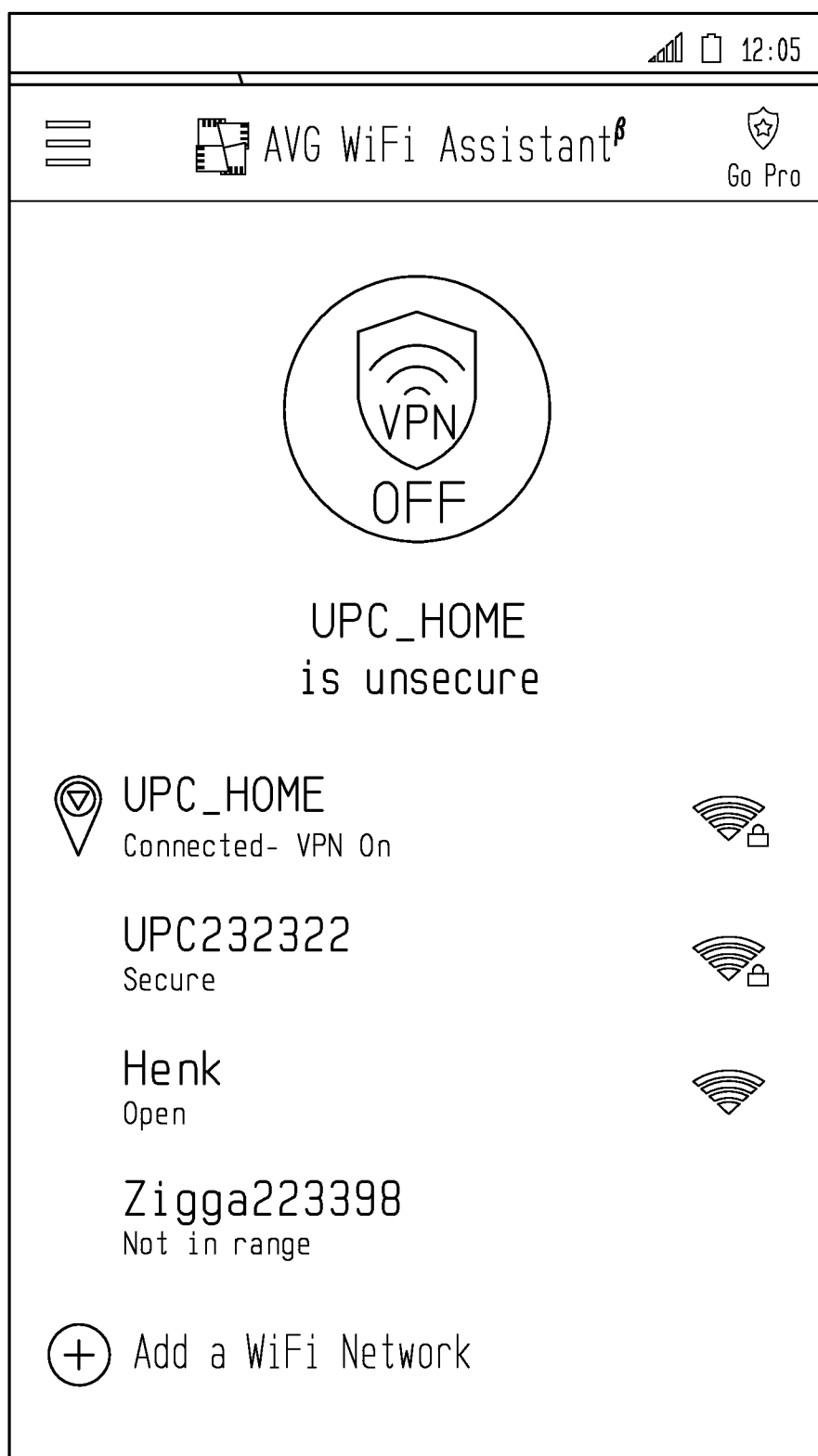


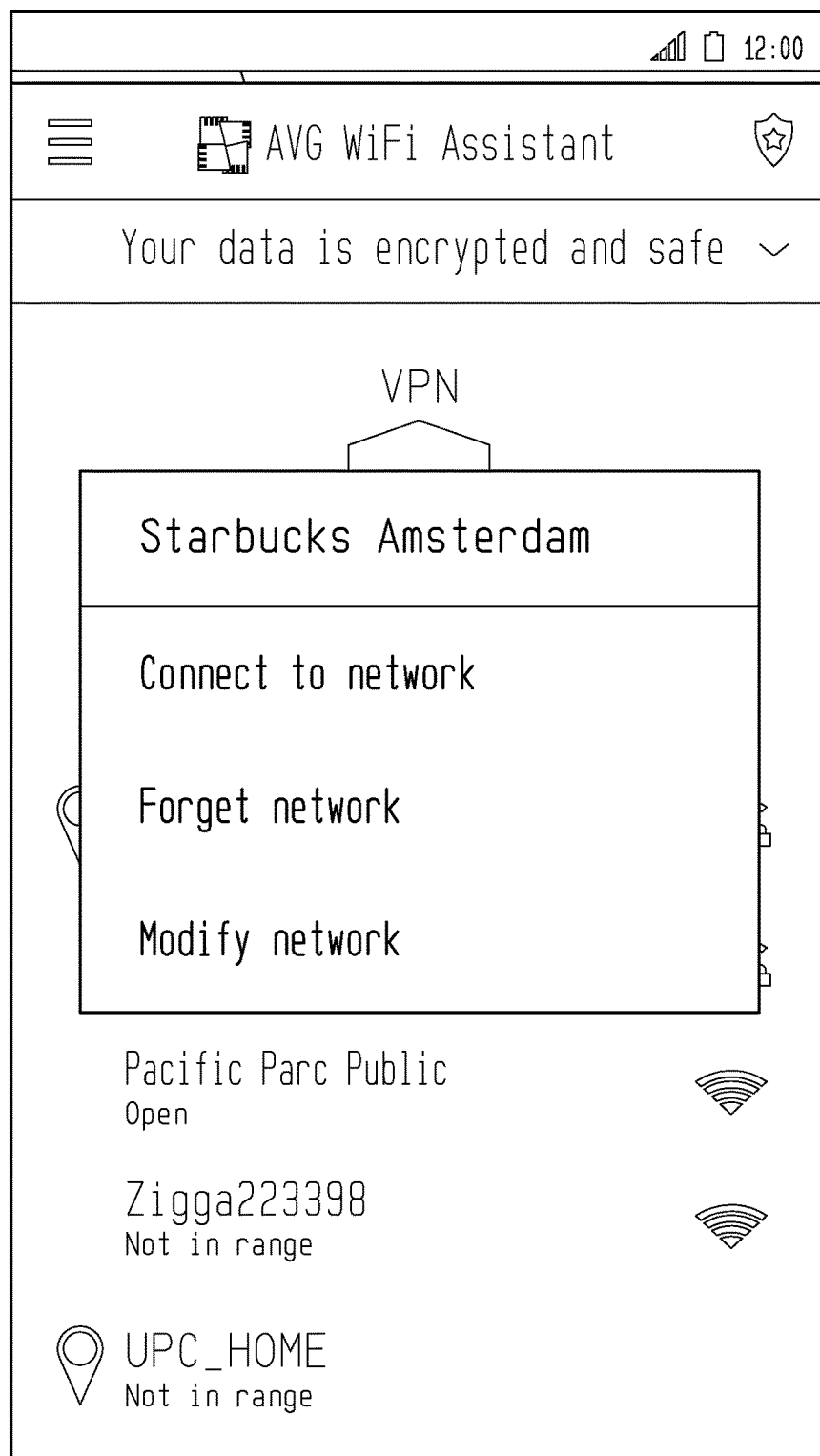
FIG. 8

FIG. 9

FIG. 10

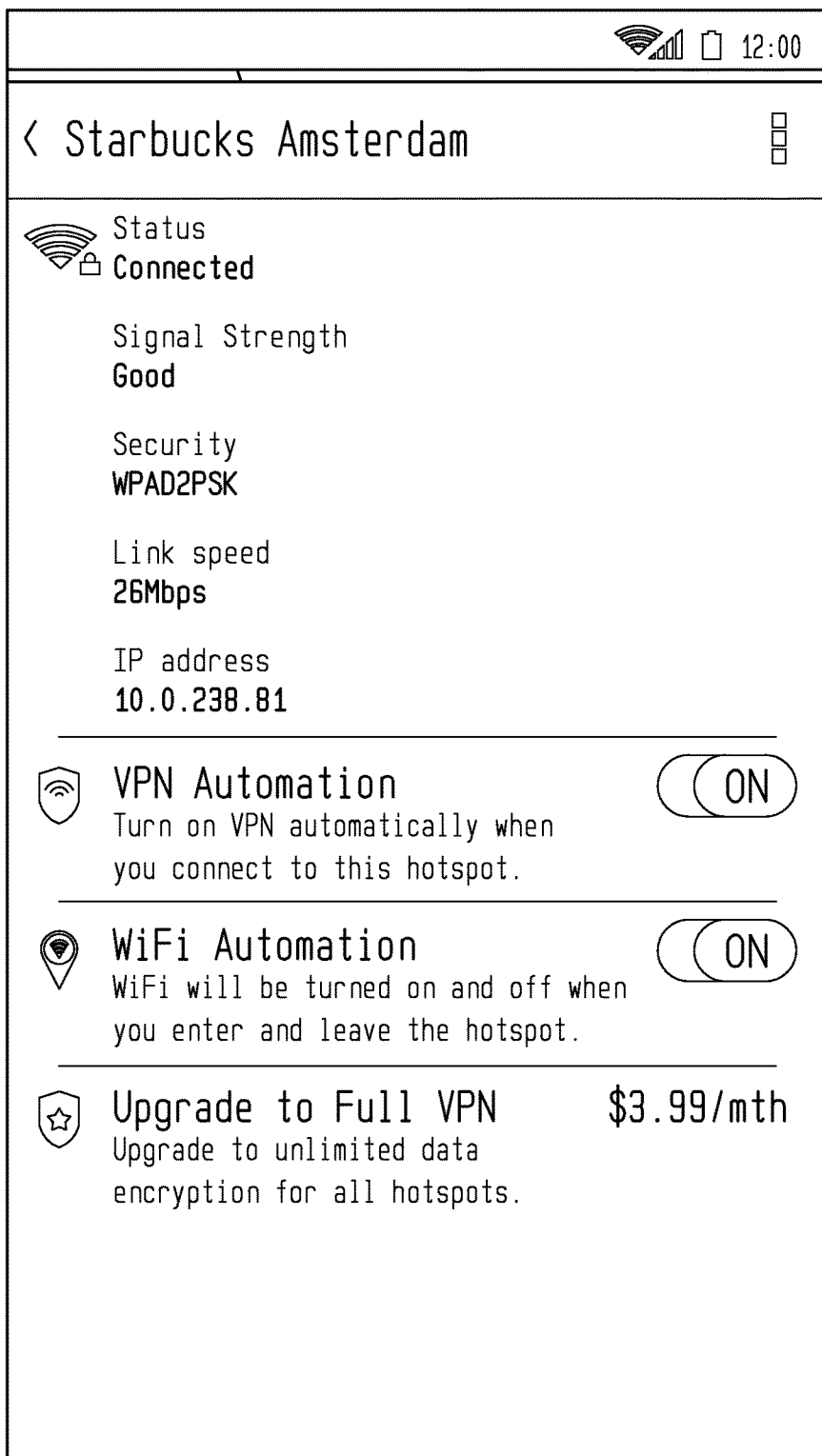


FIG. 11







   12:00	
< UPC_HOME 	
WiFi Automation	
	WiFi turns ON when you enter a hotspot <input checked="" type="checkbox"/>
	WiFi turns OFF when you leave a hotspot <input checked="" type="checkbox"/>
Cell Tower ID	<input type="button" value="Clear ID"/>
1060-6964216 WCDMA (Registered)	
1060-6947351 GSM	
1060-6958974 WCDMA	
1060-6954144 WCDMA	
1060-6952114 WCDMA	
1060-6964216 WCDMA (Registered)	

FIG. 12





<div><div><div>12:00</div></div><div><div>12:00</div></div></div>	
<div><div><div>Pro</div></div><div><div>Pro</div></div></div>	
<div><div><div>• Unlimited Data Security</div><div>• Secure any number of hotspots</div></div><div>Secure Me</div><div>only \$2.99/ month</div></div>	<div><div><div>You are subscribed to AVG WiFi Assistant Pro for \$2.99/ month</div><div><div>• Unlimited Data Security</div><div>• Secure any number of hotspots</div></div><div>Cancel Subscription</div></div></div>
<div><div>Data Secured</div><div>500 MB/ 15 days left</div><div></div><div>Secure up to 500MB data/ month for Free</div><div>Hotspot Secured 0 of 5</div><div>Secure up to 5 hotspots for Free Powered by SurfEasy</div></div>	<div><div>Hotspots Secured</div><div>UPC HOME</div><div>Office</div><div>Secure up to 5 hotspots for Free Powered by SurfEasy</div></div>

FIG. 13

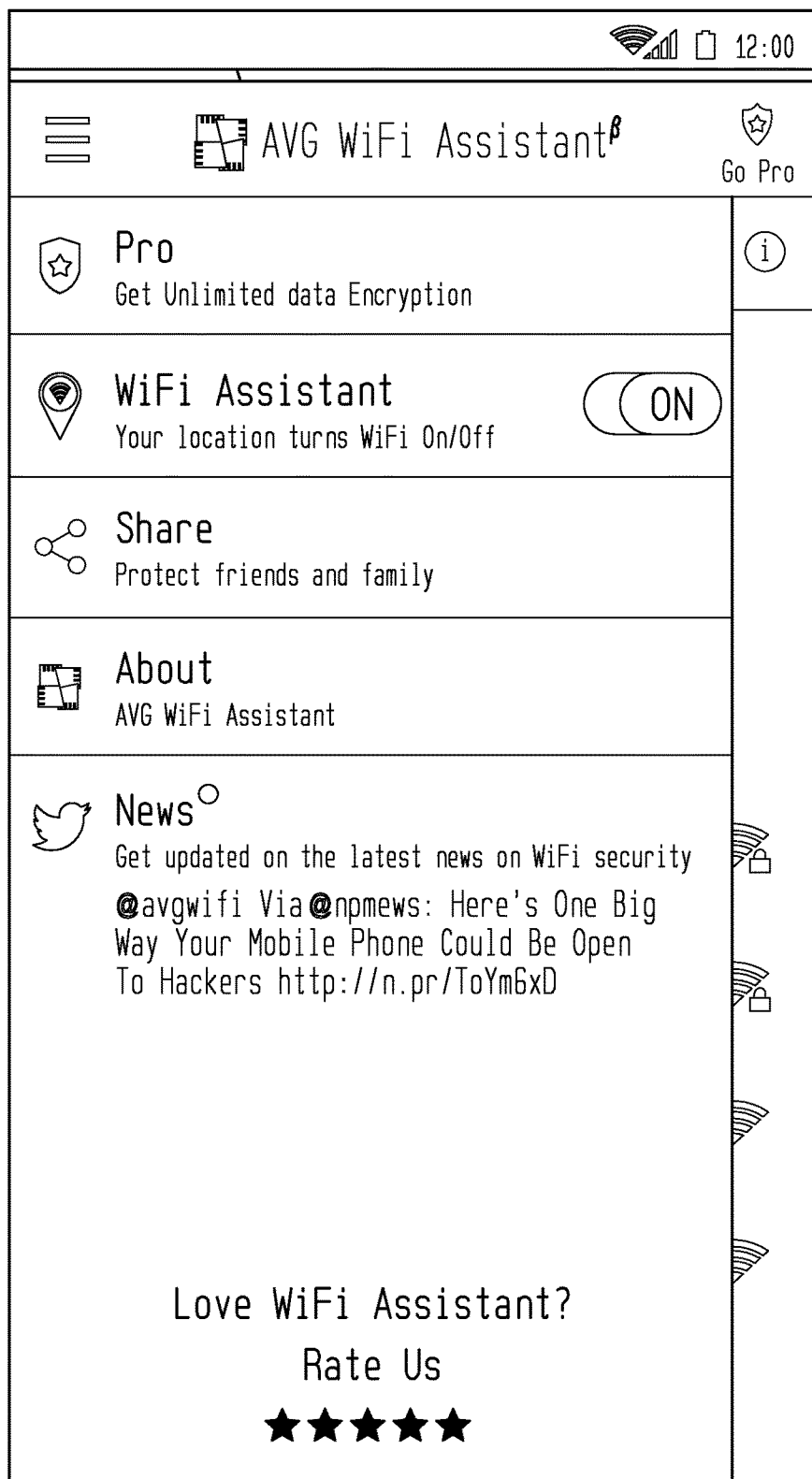


FIG. 14

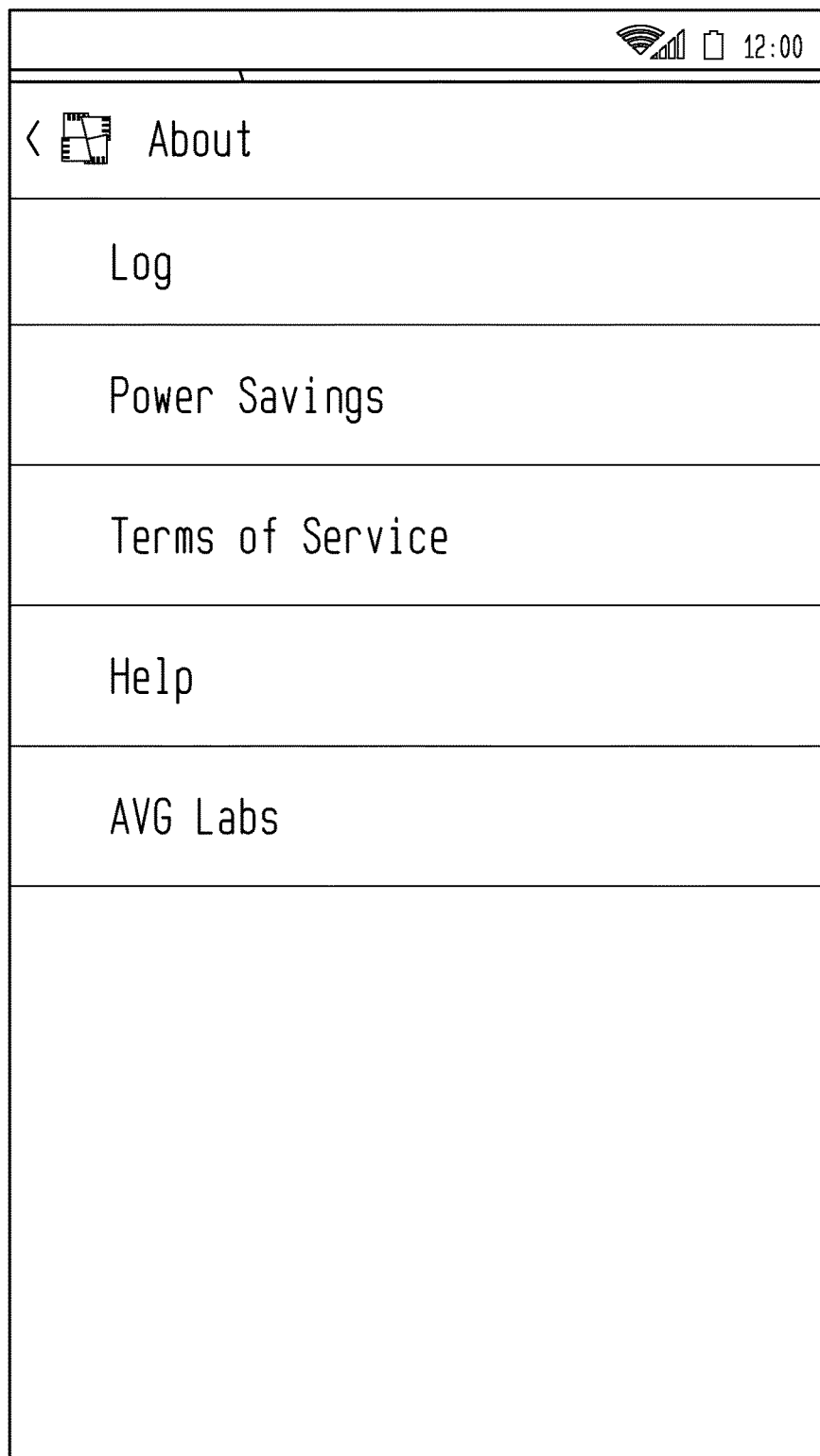
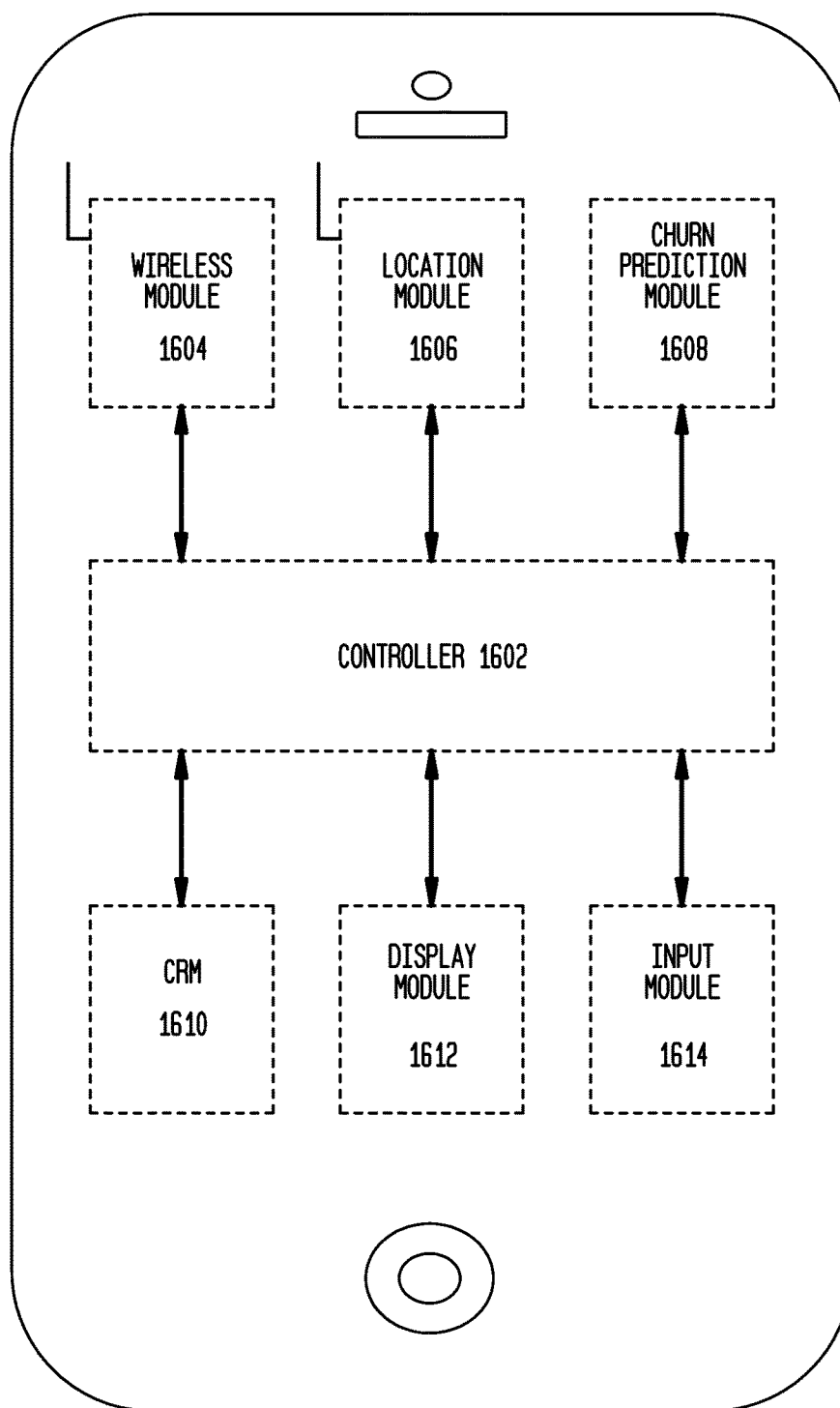


FIG. 15

ReferralSource	Country	Pro	LimitBandwidth	LimitHotspots	Language	OSVersion	City	Brand	Model	Type	ScreenRes1	ScreenRes2
WiFiAssistant	Hong Kong	1 Free VPN	1 Not reached	1 Not reached	zh-tw	4.3	Hong Kong	Sony	C5306	mobile	1184	720
(direct)	Mexico	1 Free VPN	1 Not reached	1 Not reached	es-us	4.3	Ciudad Nezahualcoyotl	Huawei	Y530-U051	mobile	854	480
(direct)	United Kingdom	1 Free VPN	2 Limit reached	1 Not reached	en-gb	5	Norwich	Samsung	SM-N900S	mobile	1920	1080
WiFiAssistant	United Kingdom	1 Free VPN	1 Not reached	1 Not reached	pl-pl	5	Stalowa Wola	LG	D855	mobile	2392	1440
(direct)	Canada	1 Free VPN	2 Limit reached	1 Not reached	en-ca	4.4.2	Montreal	Samsung	SGH-I747M	mobile	1280	720
WiFiAssistant	United States	1 Free VPN	1 Not reached	1 Not reached	en-us	4.4.4	Clarksville	Unknown	Model	NULL	NULL	NULL
(direct)	Lithuania	1 Free VPN	2 Limit reached	1 Not reached	en-gb	4.4.4	Panevezys	Sony	D5803	mobile	1184	720
WiFiAssistant	Guadeloupe	1 Free VPN	1 Not reached	1 Not reached	fr-fr	4.4.2	NULL	Unknown	Model	NULL	NULL	NULL
(direct)	Mexico	0 No VPN	0 No VPN	0 No VPN	es-us	4.4.2	Merida	Samsung	SGH-M919V	mobile	1920	1080
WiFiAssistant	Germany	0 No VPN	0 No VPN	0 No VPN	de-de	5	Berlin	Samsung	SM-G300F	mobile	1920	1080
(direct)	Germany	0 No VPN	0 No VPN	0 No VPN	de-de	4.4.2	Berlin	Samsung	GT-I9505	mobile	1920	1080
(direct)	United Kingdom	1 Free VPN	1 Not reached	1 Not reached	en-gb	4.4.4	Torquay	Samsung	SM-N910F	mobile	2560	1440
WiFiAssistant	Italy	0 No VPN	0 No VPN	0 No VPN	it-it	5	Macerata	Samsung	SM-G300F	mobile	1920	1080
WiFiAssistant	Taiwan	0 No VPN	0 No VPN	0 No VPN	zh-tw	4.4.4	NULL	Sony	D6653	mobile	1776	1080
WiFiAssistant	Spain	1 Free VPN	2 Limit reached	1 Not reached	es-es	4.4.4	Madrid	Samsung	SM-N910F	mobile	2560	1440
WiFiAssistant	France	0 No VPN	0 No VPN	0 No VPN	fr-fr	4.4.2	Paris	Samsung	GT-I919S	mobile	960	540
(direct)	India	1 Free VPN	1 Not reached	1 Not reached	en-in	5.0.2	Chennai	Motorola	XT1068	mobile	1184	720
WiFiAssistant	Canada	1 Free VPN	1 Not reached	1 Not reached	en-ca	4.4.4	Toronto	Sony	D6603	mobile	1776	1080
(direct)	Italy	1 Free VPN	2 Limit reached	1 Not reached	it-it	4.1.2	Milan	Samsung	GT-I8130	mobile	800	480
(direct)	Singapore	1 Free VPN	1 Not reached	1 Not reached	en-ch	4.0.3	Lahore	HTC	X515	mobile	960	540
(direct)	Sweden	1 Free VPN	1 Not reached	1 Not reached	sv-se	4.4.4	Mjølby	Sony	D5503	mobile	1184	720
(direct)	Germany	1 Free VPN	1 Not reached	1 Not reached	de-de	5	Cologne	Samsung	SM-G300F	mobile	1920	1080

FIG. 16
MOBILE DEVICE 1601



PREDICTING CHURN FOR (MOBILE) APP USAGE

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] The present application claims the benefit of U.S. Provisional Patent Application No. 62/265,552, filed on Dec. 10, 2015, the disclosure of which is hereby incorporated herein by reference as if fully set forth.

FIELD OF THE INVENTION

[0002] The present invention generally relates to personal electronic devices, such as smartphones, tablets and computers, and in particular to metrics and models for predicting what percentage of the population that tries a given application on a user device will continue using it at various subsequent time periods, and with what regularity.

BACKGROUND OF THE INVENTION

[0003] This application relates generally to application software, commonly referred to as an “app.” Apps are computer software designed to help the user to perform specific tasks. Apps may be executed on a variety of computing devices, such as on mobile devices including smartphones. For example, mobile apps are software applications designed to run on smartphones, tablet computers and other mobile devices. The apps are available through application distribution platforms, which are typically operated by the owner of the mobile operating system, such as, for example, the Apple App Store, Google Play, Windows Phone Store and BlackBerry App World. Mobile devices, such as smartphones and tablet computers, are designed to readily accept the apps for installation and operation.

[0004] Churn prediction is the process of determining the percentage of the population (i.e., users) that will stop using a given service or product after having initially tried it. All mobile app publishing companies are interested in the percentage of users that will be still using an app one week after a user has installed the app.

[0005] Traditionally, churn prediction is based upon a consumer’s characteristics, such as age, sex, and zip code. However, these categories are simply too broad to granularly predict how a user of an app seems to like using it, and whether he or she will continue to use it, continue to experiment with it, or decide that it is not useful to him or her.

[0006] Thus, it is often how a user interacts with a new application that both determines whether they will continue, and if they have a sufficient interest level to actually learn how to use it in a meaningful way. This is not just a matter of age, sex and residential neighborhood of a user. It is more about their personality, their needs, and how they relate to a given software application during that window of time in which the application is making its “first impression” on them.

[0007] What is needed in the art are methods for more accurately predicting churn based upon actual user activity involving the application.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] FIG. 1 is a chart of exemplary discriminating values and associated discriminating patterns for a churn class according to an exemplary embodiment of the present invention;

[0009] FIG. 2 is a chart of exemplary discriminating patterns for a loyals class according to an exemplary embodiment of the present invention;

[0010] FIGS. 3-14 are exemplary screen shots (or portions thereof) from an exemplary beta application called “AVG WiFi Assistant” used to test churn and loyalty prediction according to an exemplary embodiment of the present invention;

[0011] FIG. 3 depicts an exemplary activation screen;

[0012] FIG. 4 depicts an exemplary OnboardWiFi screen;

[0013] FIG. 5 depicts an exemplary OnboardVPN screen;

[0014] FIG. 6 depicts exemplary coaching bubbles displayed in connection with the Onboarding screen of FIG. 4, to help users to help users learn the functionality of Wifi Assistant;

[0015] FIG. 7 depicts an exemplary Home screen;

[0016] FIG. 8 depicts an exemplary screen displayed to a user upon the user choosing a WiFi hotspot;

[0017] FIG. 9 is an exemplary Secure Hotspot screen displayed to a user upon choosing a WiFi hotspot as in FIG. 8;

[0018] FIG. 10 depicts an exemplary WiFi Settings screen, used to set parameters of a chosen WiFi hotspot;

[0019] FIG. 11 depicts an exemplary WiFi Settings Advanced screen;

[0020] FIG. 12 depicts an exemplary Upgrade screen shot pair;

[0021] FIG. 13 depicts an exemplary Side Menu screen, accessed by a user from the screen shown in FIG. 7, for example;

[0022] FIG. 14 depicts an exemplary About screen;

[0023] FIG. 15 is an exemplary table containing various feature vectors according to an exemplary embodiment of the present invention; and

[0024] FIG. 16 depicts an exemplary mobile device on which a churn prediction module may be deployed according to an exemplary embodiment of the present invention.

SUMMARY OF THE INVENTION

[0025] A churn prediction model is presented that uses both behavioral data as well as user characteristics to predict whether a given user will churn (i.e., stop using) an application. Initially a training set of user interactions can be correlated to a churn probability value for various sequences of user activity. Then, as regards a real time user, user actions in navigating through the app may be recorded, and this information can be used, in addition to user characteristics, to predict the probability that this user will churn, thus implementing in a “nip churn in the bud” approach (or, the inverse, remain loyal and continue to use the app). In some embodiments, a partial set of user actions can be identified as subsequences of known churn sequences. To users performing those subsequences of activity, a real time message, offer or promotion may be sent so as to influence them not to churn. In exemplary embodiments of the present invention, user data may be uploaded from a user’s device to proprietary or cloud servers. Churn analysis, or a more detailed churn analysis, using up to the minute collective data for the given app, may, for example, be performed on those servers.

DETAILED DESCRIPTION OF THE INVENTION

[0026] In exemplary embodiments of the present invention a churn prediction model can be provided that uses both behavioral data as well as user characteristics. In particular, how a user navigates through an app can be recorded, and this data, in addition to a set of user characteristics, can be used to predict the probability that the user will churn (i.e., stop using the application). Estimating this value can be extremely valuable in multiple ways, which can include, for example:

- [0027]** 1. Allowing app developers/designers to reshape the flow of an app. That is, using exemplary embodiments of the present invention, over a period of time it can become clear at which point(s) in the app a user will, or will likely, churn. This can indicate that the user interface of the app can be improved, and even inform developers/designers as to what ways to improve it;
- [0028]** 2. Providing insights when certain actions (e.g., coupons, promotions, etc.) should be launched to keep the customer using, or engaged with, the app.

I. CONCEPTUAL DEFINITIONS

[0029] To better understand certain concepts underlying various exemplary embodiments of the present invention, definitions of the following key terms are provided:

[0030] Behavioral data refers to data regarding how a given app is being used, i.e., how a customer is interacting with the app. In particular, an ordered series of events (i.e. a sequence) that represent various different actions that can be taken by users of the app can be considered. For example, a chat app may consist of the following screens: welcome (w), signup (s), tutorial (t), address book (a), help (h), and chat (c). For a given user i , an ordered series of events may be: (w,s,c,c,c,h,a), while for another user j it may be (w,s,t,a,c,c). In general, this data is collected by enabling analytics for each event so that when a certain event occurs, it is recorded and send to a database.

[0031] More formally, we let $E=\{e_1, \dots, e_n\}$ be the set of n distinct events a user may perform, comprising an alphabet of different screen identifiers—such as, for example, $\{w, s, t, a, h, c\}$ in the exemplary chat app referred to above. Thus, the set E is the universe of all possible events (i.e. screen that can be visited and interactions that may be performed at each screen). We then let a sequence $Y=\langle y_1, \dots, y_n \rangle$ be an ordered list of events actually performed by the user, i.e. $y_i \in E$ for $1 \leq i \leq m$. For behavioral data, a transaction can be represented by a tuple (an ordered set of values) (u_i, Y) , where u_i is a user identifier and Y a sequence of events. Thus, a tuple associates a given user u_i with a given interactive sequence of events.

[0032] Customer characteristic data refers to all personal and demographic data that can be obtained about a user. For example, age, sex, country of residence, income, phone brand/model, version of operating system, whether a user is using the pro version (or other version type or designator), etc. Available customer characteristic data varies for different apps, and is mainly dependent upon which data is collected by the app. For example, for subscription based apps, payment information and email address will be present; on the other hand, for dating applications sex, age and city or town of residence is generally always known.

[0033] More formally, for every customer there is a binary vector of length p : $X=\{x_1, \dots, x_p\}$. Where each x_i , for $1 \leq i \leq p$ represents the presence/absence of the binarized feature i . As with the behavioral data, a transaction is a tuple (u_i, X) , where u_i is a user identifier and X a feature vector. Here a tuple associates a given user with a given set of customer characteristic features. FIG. 15 is a table containing various exemplary feature vectors. As can be seen with reference thereto, each row of the table contains data from a different customer or user of the exemplary WiFi Assistant application described below in connection with FIGS. 3-14. The data includes the user's country, city and country, the language they used, and information regarding their user device, such as brand, model, type, and the operating system it used.

II. CHURN PREDICTIONS: A TWO-STAGE PROCESS

[0034] In exemplary embodiments of the present invention a churn prediction application may consist of two stages:

- [0035]** 1. A learning stage where a churn model is learned, derived or generated from the data—based on historical data a statistical model to predict churn is learned or generated.
- [0036]** 2. A deployment stage, where, in real time settings, a churn model is used to predict whether a user is likely to churn. After some period of time, the model may then be retrained with more recent data.

[0037] In exemplary embodiments of the present invention, the entire analysis can be run, with the now larger data set including the more recent data to find a new list of discriminating patterns. However, in some embodiments, after some time it can also be beneficial to only use the newer data.

[0038] These stages are next described.

[0039] A. Learning the Churn Model

1. INTRODUCTION AND DEFINITIONS

[0040] In order to derive or learn a statistical model for churn prediction from interaction data, multiple historical data transactions are needed. That is, behavioral as well as customer characteristics data from multiple customers is required. Moreover, for this “learning set” it is known whether a customer churned or not, so the data can be correlated to a known outcome.

[0041] Let $D=\{(u_1, Y_1), \dots, (u_n, Y_n)\}$ be the transaction database of behavioral characteristics data, and $A=\{(u_1, X_1), \dots, (u_n, X_n)\}$ be the transaction database of user characteristics data. As noted above, the elements of each such transaction database is a set of tuples. Similarly, $U=\{u_1, \dots, u_n\}$ is the database of users. Furthermore, in the training data, because it is historical, a binary class label (for example: “churned” or “loyal”) is also known for each user. That is, there is a known labeling function $C: U \rightarrow \text{churned/loyal}$ for all users in the historical data set. With D^{churn} , D^{loyal} we denote the subset of users in D that respectively have churned/are loyal, i.e., $D^{\text{churn}}=\{(u_i, Y_i) | (u_i, Y_i) \in D \text{ and } C(u_i)=\text{churn}\}$. Moreover, we need the notion of corresponding transactions between the behavioral transactional database and characteristics transaction database,

$$\text{for } D' \subseteq D, A[D'] = \{(u_i, X) | (u_i, X) \in A \text{ and } \exists (u_i, T) \in D' \text{ where } u_i = u'_i\}.$$

[0042] Intuitively, $A[D']$ contains all the transactions of A, where the user identifier is both in A and D' .

[0043] As a result, the triple $(Y_i, X_i, C(u_i))$ provides the behavioral and characteristics data for user u_i , and also indicates whether the user has churned.

[0044] The overall goal of churn prediction is to learn the probability of $(Y_i, X_i, C(u_i)) = \text{churned}$, i.e., to learn the probability that the user u_i with behavioral and characteristics data X_i and Y_i will churn.

2. COMMON APPROACH FOR CHURN PREDICTION

[0045] Common practice for learning a statistical model for outcome prediction is to use Logistics Regression, Naïve Bayes, or other predictive models on input sets A and the class label mapping C. More precisely, in exemplary embodiments of the present invention, a churn prediction model M may be derived by using an off-the-shelf predictive model ϕ applied on A and their corresponding class labels, i.e., $M = \Phi(A, C)$. For example, with Naïve Bayes the probability of churning for user j—i.e.,

$$P(\text{churn} | (u_j, X_j)) = \frac{P(\text{churn})P(X_j | \text{churn})}{P(X_j)},$$

[0046] where $P(\text{churn})$ is the prior probability of churning on the training dataset, and $P(X_j | \text{churn})$ can be estimated by using a Maximum Likelihood estimate on A.

3. TAKING BEHAVIORAL DATA INTO ACCOUNT

[0047] Intuitively, the group of users can be first split into multiple (possibly partially overlapping) groups. The splitting can be done, for example, based on how customers are using the app, i.e., users with similar behavior are grouped together. Consequently a statistical model for churn prediction can be separately learned for each group.

[0048] In exemplary tests run by the inventors, the grouping of the users was done based upon the behavioral characteristics of a user. In particular, users with similar behavior were grouped together.

[0049] It is noted that, as used above, the term or concept “similar behavior” is used in the sense of similar browsing behavior, i.e., how a user navigated through the application. Frequent sequential patterns is one way, for example, to capture the similarity; when enough people visited screens in the sequence A-B-C, this can be identified as a frequent pattern. The grouping may then be further done on only the frequent patterns that are actually discriminating, because the non-discriminating ones are not informative for prediction purposes.

[0050] Additionally, the grouping was optimized to select those groups with deviating overall churn probability, that is, groups where the overall churn probability was far higher/lower than the overall churn probability. In order to compute this grouping, well known techniques such as, for example, frequent sequence mining, hidden Markov models, or variable length Markov chains can be used.

[0051] Next described in detail is how this deviating overall churn probability can be obtained using frequent sequence mining techniques.

4. FINDING FREQUENT SEQUENTIAL DISCRIMINATING PATTERNS

[0052] For two sequences $Y = \{y_1, \dots, y_m\}$, $Z = \{z_1, \dots, z_k\}$ it is said that Z is a subsequence of Y, denoted as $Z \leq Y$, if and only if: there exists a sequence of length k in Y such that:

$$y_i = z_1, \dots, y_{i+k} = z_k \text{ with } 1 \leq i \leq m-k.$$

[0053] The cover of a sequence Z in the transaction database D consists of the bin or cluster of transactions that supports Z in D:

$$\text{cover}(Z, D) = \{(u_i, Y) | (u_i, Y) \in D, Z \leq Y\}.$$

[0054] The support of a sequence Z in transaction database D is the number of transactions in the cover of Z in D, $\text{supp}(Z, D) = |\text{cover}(Z, D)|$. Frequent sequence mining is about deriving all sequences with a support larger than a user supplied minimum support threshold. Discriminating frequent sequential patterns are patterns that discriminate between the classes, i.e., patterns that are far more common in the churned group than in the loyal customer group. Formally, the discriminatory values of a frequent sequential pattern P equals

$$\max(\text{supp}((P, D^{\text{churned}}) / \text{supp}(P, D^{\text{loyal}}), \text{supp}((P, D^{\text{loyal}}) / \text{supp}(P, D^{\text{churned}}))).$$

[0055] In general the discriminatory value is a user defined parameter and can be used to fine tune the algorithm. An exemplary discriminatory value can be 0.6.

[0056] Given the previous definitions and discussion, pseudo-code for learning of the churn model is next presented. The output of the algorithms is the set of all frequent discriminating patterns \mathcal{P} and a set of models \mathcal{M} .

[0057] 1. Learn Churn Model (A, D, C)

[0058] 1.1. $\mathcal{M} = \{ \}$ // initialize the set of churn models to be empty

[0059] 1.2. $\mathcal{P} = \text{find discriminating frequent patterns on } (D)$

[0060] 1.3. For each frequent sequential pattern P in \mathcal{P}

[0061] 1.3.1. $\mathcal{M}_P = \text{learn Churn Model on } A[\text{cover}(P, D)]$

[0062] 1.3.2. $\mathcal{M} = \mathcal{M} \cup \mathcal{M}_P$

[0063] 1.4. Return $(\mathcal{M}, \mathcal{P})$

[0064] In exemplary embodiments of the present invention, a set of models arises because for every pattern, a classification model is constructed. A pattern is supported by a certain part of the data, and it is on this part the associated model may then be constructed.

[0065] B. Deploying the Churn Model

[0066] In order to predict whether user u_j will churn, it is assumed that the customer characteristics of u_j , i.e., X_j and part of the behavioral characteristics, that is Y_j , are available. The behavioral characteristics of u_j are then used to determine a group with a similar behavioral profile; this group has some pattern P as a common descriptor.

[0067] For example, from the behavioral characteristics Y_j , we can, for example, first find the best match from the set of patterns P, where best means in the sense of the longest subsequence. Thus, for all p in P, we test if p is a subsequence of Y_j . From all the matches, we then, for example, select the longest. In case there are multiple maximal subsequences that match, we select the most discriminating. If the unlikely case occurs that there are again multiple

maximal subsequences, all equivalently discriminating, we may use both (and average the result).

[0068] Next, the model trained for pattern P, i.e., M_p can be used to determine the churn probability of X_j . It is noted that since Y_j is changing over time, the predictive model that will be selected can also change over time.

[0069] Practically, for all patterns that have been discovered in the learning phase, it is desired to find the best match with the current behavioral characteristics sequence Y_j . For the “best”, a simple heuristic may be used: the longest sequence that is a subsequence of Y_j is the best fit. When multiple equally long sequences are suited, we can compute for each of them a predicting score. Then the largest score for the churn class and the largest score for the loyal class are selected and returned. In terms of pseudo-code:

```
[0070] 1. Predict Churn ( $X_j, Y_j, \mathcal{M}, \mathcal{P}$ )
[0071] 1.1. maxmatch=0
[0072] 1.2. Out={ }
[0073] 1.3. For all P in  $\mathcal{P}$  (#start with longest patterns first)
[0074] 1.3.1. If (( $P \leq X_j$ ) and ( $|P| \geq \text{maxmatch}$ ))
[0075] 1.3.1.1. Out=Out $\cup$ {Compute probability of churn and loyal ( $M_p, Y_j$ )}
[0076] 1.3.1.2. maxmatch=IP
[0077] 1.4. (churn,loyal)=maximum value from Out
[0078] 1.5. Return (churn,loyal)
```

[0079] Besides predicting whether a user will churn, another aspect of exemplary embodiments of the present invention is to influence users to visit certain screens, i.e. to change user behavior with respect to app usage. Thus, we want to change how the user is using the app based upon the insights we have gained from the model as applied to his or her behavior thus far. For example, given that we have detected two groups of users of the chat app described above, and the first group has the behavioral pattern (w (welcome), s (signup), c (chat)) with overall probability of churn for this group equal to 90%. The second group, on the other hand, has the behavioral pattern (w (welcome), s (signup), t (tutorial)) with an overall churning probability of 10%. So, if a user visited the stages w,s we want to influence the user to now visit the tutorial screen t. This influencing can be done both online as well as offline. Offline (i.e., beforehand) influencing can, for example, be done by changing the User Interface to make the tutorial screen more visible. Online influencing (meaning in real time) can for example be achieved by automatically sending promotions.

[0080] Thus, a churn prediction approach is provided based on app usage data in addition to customer characteristics.

III. EXAMPLES

[0081] A. Example of Discriminating Patterns

[0082] Churned Patterns

[0083] FIG. 1 depicts examples of discriminating patterns for the Churn class. The discriminating value equals the probability that the sequence is from the “Churn” class.

[0084] Loyal Patterns

[0085] FIG. 2 depicts examples of discriminating patterns for the Loyals class, the discriminating value equals the probability that the sequence is from the “Loyal” class.

[0086] As may be readily noted, the discriminating value, Loyal=[1–discriminating value, Churn].

[0087] B. Detailed Churn Prediction Examples Next described are two examples of churn prediction according to exemplary embodiments of the present invention.

Example 1

[0088] Taking the following behavioral data, from the group of churned users:

```
[0089] Networkinfo|GetStarted Pressed 0|OnboardWifi
Start yes|OnboardVPN yes 0|Bubble HotspotAutomation
Display|Bubble VPN Display|Bubble HotspotAutomation
Dismissed|Bubble VPN Dismissed|Home
Button AddWifiNetwork|SecureHotspot yes 0|SecureHotspot
ContinueNoVPN 0|Home Button AddWifiNetwork|Home
Button AddWifiNetwork|Home Button AddWifiNetwork|Home
Button SideMenu|Home Button SideMenu|Side WifiAssistant
Off|Side WifiAssistant On|Home Button AddWifiNetwork|Home
Button AddWifiNetwork|Home Button Upgrade|Upgrade
SecuredHotspot WifiSettings
```

[0090] Initially, when only the first action of a user is known, i.e., as in this case, it is only known thus far that the Networkinfo screen had been visited, there is no pattern that matches this sequence yet. As a result, the churn predicting model over the whole dataset (i.e. the global model) is used. The outcome is that this user will churn with a probability of 18.93% and hence be loyal with a probability of 81.07%.

[0091] It is noted that the global churn model is the churn model that is derived by using all of the available data. As above, this model may be used as fallback when there is no match with any of the derived discriminating sequential patterns.

[0092] As more screens were visited by the user, the outcome did not change until 14 screens were recorded. Here is a short trace of some of the intermediate stages:

```
[0093] Networkinfo|GetStarted Pressed 0=>no match,
global model P(churn)=18.93%
```

```
[0094] Networkinfo|GetStarted Pressed 0|OnboardWifi
Start yes|OnboardVPN yes 0=>no match, global model
P(churn)=18.93%
```

[0095] This is because none of the discriminating patterns we derived matched (partly) with the first 13 screens. However, after the first 13 screen we have the following input:

```
[0096] Networkinfo|GetStarted Pressed 0|OnboardWifi
Start yes|OnboardVPN yes 0|Bubble HotspotAutomation
Display|Bubble VPN Display|Bubble HotspotAutomation
Dismissed|Bubble VPN Dismissed|Home
Button AddWifiNetwork|SecureHotspot yes 0|SecureHotspot
ContinueNoVPN 0|Home Button AddWifiNetwork|Home
Button AddWifiNetwork|Home Button AddWifiNetwork|Home
Button SideMenu|
```

[0097] There was now a match with one of the discriminating patterns from the churn class:

```
[0098] Home Button AddWifiNetwork|Home Button
AddWifiNetwork|Home Button SideMenu
```

[0099] By selecting the churn predictive model trained only on data instances that supported this rule, we got: P(churn)=100%.

[0100] The rest of the screens that the user added did not lead to any different patterns matching the input sequence, and hence the best matching rule for this user remained:

```
[0101] Home Button AddWifiNetwork|Home Button
AddWifiNetwork|Home Button SideMenu
```

Example 2

[0102] The second example describes the various screens that a loyal user visited:

[0103] networkinfo|GetStarted Pressed 0|OnboardWifi Start yes|Bubble HotspotAutomation Display|Bubble VPN Display|Bubble HotspotAutomation Dismissed-|Bubble VPN Dismissed|SecureHotspot AlwaysOn Checked|SecureHotspot yes 0|SecureHotspot ContinueNoVPN 0|Home Button SideMenu|Home VPN On

[0104] As in Example 1, here as well there were no matching discriminating patterns after the first three screens were visited. Thus, the following initial churn prediction was made:

[0105] networkinfo|GetStarted Pressed 0|OnboardWifi Start yes=>

[0106] no match, global model $P(\text{churn})=28.36\%$.

[0107] With the addition of the next screen a match was found with the pattern:

[0108] OnboardWifi Start yes|OnboardVPN yes 0|Bubble HotspotAutomation Display

[0109] and the corresponding model estimates $P(\text{churn})=0.04\%$. Adding another screen results in two matching discriminating patterns: the previous pattern “OnboardWifi Start yes|OnboardVPN yes 0|Bubble HotspotAutomation Display” and the following pattern:

[0110] OnboardWifi Start yes|Bubble HotspotAutomation Display|Bubble VPN Display.

[0111] Since the latter is the longer pattern (three events versus two), as per the rule described above, this latter one was selected and used for prediction, and thus now $P(\text{churn})=0.06\%$.

IV. EXEMPLARY APPLICATION FLOW: SCREENS AND THEIR CORRESPONDING NAMES

[0112] Next described is an exemplary application flow, with reference to FIGS. 3-14. In the following description, events are sent to Google Analytics (“GA”). Each event sent to Google Analytics is an event in our behavioral database. In exemplary embodiments, other mobile tracking solutions might be used, such as, for example, Adobe Omniture, Flurry Analytics or any in-house developed software that tracks and sends app user interactions. In exemplary embodiments of the present invention, user data may be uploaded from a user’s device to proprietary or cloud servers. Churn analysis, or a more detailed churn analysis, using up to the minute collective data for the given app, may, for example, be performed on those servers.

[0113] An exemplary application flow is described with reference to the screen shots of FIGS. 3-14. With respect to each screen, a list of user interactions is provided, and the data based on such interactions that is sent to an analytics environment, such as GA, for example, is provided. Each time a user accesses a screen, or interacts with it, data capturing the user’s actions can be sent to GA, as noted below.

[0114] Activation

[0115] When user opens the screen shown in FIG. 3, we send to GA the name of the page he is looking at, on this case:

[0116] Activation

[0117] When the user presses the “GetStarted” button the information sent to GA is:

[0118] “GetStarted”, “Pressed_0”, “EventCounter”, null

[0119] “NetworkInfo”, “mobile operator type_mobile operator name”, “EventCounter”, null

[0120] where the mobile operator type can, for example, contain the following information:

[0121] GSM or CDMA or None or SIP

[0122] OnboardWifi

[0123] FIG. 4 depicts an exemplary Onboarding WiFi screen. When a user sees this screen the name of the page he is looking at is sent to GA, in this case:

[0124] OnboardWifi

[0125] When the user taps “Start WiFi Automation” in FIG. 4, the information sent is:

[0126] “OnboardWifi”, “Start_Yes”, “EventCounter”, null

[0127] When the user taps “Maybe later” in FIG. 4, the information sent is:

[0128] “OnboardWifi”, “Start_Later”, “EventCounter”, null

[0129] OnboardVPN

[0130] FIG. 5 depicts an exemplary Onboarding VPN screen. When a user sees this screen we send the name of the page he is looking at to GA, in this case:

[0131] OnboardVPN

[0132] When the user taps “Got it”, the information sent to GA is:

[0133] “OnboardVPN”, “Yes_0”, “EventCounter”, null

[0134] Coaching Bubbles

[0135] As part of the Onboarding process shown in FIGS. 4 and 5, coaching bubbles can, for example, be displayed, to help users learn the functionality of WiFi Assistant.

[0136] Exemplary coaching bubbles are shown in FIG. 6.

[0137] When the user sees these bubbles, the information sent is:

[0138] “Bubble”, “HotspotAutomation_Display”, “EventCounter”, null

[0139] “Bubble”, “VPN_Displayed”, “EventCounter”, null

[0140] “Bubble”, “AssistantOff_Display”, “EventCounter”, null

[0141] When the user dismisses these bubbles, the information sent is:

[0142] “Bubble”, “HotspotAutomation_Dismissed”, “EventCounter”, null

[0143] “Bubble”, “VPN_Dismissed”, “EventCounter”, null

[0144] “Bubble”, “AssistantOff_Dismissed”, “EventCounter”, null

[0145] Home Screen

[0146] FIG. 7 depicts an exemplary home screen. When a user opens this screen the name of the page he is looking at is sent to GA, in this case:

[0147] Home

[0148] When the user taps the “Wifi is Off”, the information sent is:

[0149] “Home”, “Wifi_On”, “EventCounter”, null

- [0150] When the user taps the “Wifi is On”, the information sent is:
- [0151] “Home”, “Wifi_Off”, “EventCounter”, null
- [0152] When the user turns on VPN, the information sent is:
- [0153] “Home”, “VPN_On”, “EventCounter”, null
- [0154] (not sent first time user connect to hotspot and SecureHotspot screen is displayed)
- [0155] When the user turns Off VPN, the information sent is:
- [0156] “Home”, “VPN_Off”, “EventCounter”, null
- [0157] The buttons “Wifi is Off” and “Wifi is On” are tapable, thus allowing the interaction data to be created.
- [0158] From a home screen as shown in FIG. 7, a user may tap the icon shown at the top left, seen in FIG. 7 as three horizontal bars.
- [0159] When the user taps the top left icon to access the side menu the information sent is:
- [0160] “Home”, “Button_SideMenu”, “EventCounter”, null
- [0161] When the user taps the “Go Pro”, the information sent is:
- [0162] “Home”, “Button_Upgrade”, “EventCounter”, null
- [0163] When the user taps the “+”, the information sent is:
- [0164] “Home”, “Button_AddWifiNetwork”, “EventCounter”, null
- [0165] Hotspot Interactions
- [0166] FIG. 8 depicts an exemplary Hotspot Connect screen. A user would see this screen if, for example, she entered a Starbucks in Amsterdam, The Netherlands.
- [0167] When the user taps on a hotspot and selects “Connect to network”, the information sent is:
- [0168] “Home”, “Hotspot_Connect”, “EventCounter”, null
- [0169] When the user taps on a hotspot and selects “Forget network”, the information sent is:
- [0170] “Home”, “Hotspot_Forget”, “EventCounter”, null
- [0171] When the user taps on a hotspot and selects “Modify network”, the information sent is:
- [0172] “Home”, “Hotspot_WifiSettings”, “EventCounter”, null
- [0173] When the user taps on a hotspot and selects “Disconnect to network”, the information sent is:
- [0174] “Home”, “Hotspot_Disconnect”, “EventCounter”, null
- [0175] Secure Hotspot FIG. 9 depicts an exemplary Secure Hotspot screen. A user would see this screen if, for example, this is the first time the user chose to connect to the hotspot (e.g., Starbucks Amsterdam). When this screen is displayed the name of the page the user is viewing is sent to GA, in this case:
- [0176] SecureHotspot
- [0177] When the user taps the “Secure this Hotspot” button, the information sent is:
- [0178] “SecureHotspot”, “Yes”, “EventCounter”, null
- [0179] When the user unchecks “Always use VPN for this hotspot” box, the information sent is:
- [0180] “SecureHotspot”, “AlwaysOn_Unchecked”, “EventCounter”, null
- [0181] When the user checks “Always use VPN for this hotspot” box, the information sent is:
- [0182] “SecureHotspot”, “AlwaysOn_Checked”, “EventCounter”, null
- [0183] When the user taps the “Continue without VPN” button, the information sent is:
- [0184] “SecureHotspot”, “ContinueNoVPN”, “EventCounter”, null
- [0185] WiFi Settings FIG. 10 depicts an exemplary WiFi settings screen shot. When the user opens this screen the name of the page he is looking at is sent to GA, in this case:
- [0186] WifiSettings
- [0187] User interactions with WiFi settings:
- [0188] When the user changes Wifi automation to OFF, information sent is:
- [0189] “WifiSettings”, “WifiAuto_Off”, “EventCounter”, null
- [0190] When the user changes Wifi automation to ON, information sent is:
- [0191] “WifiSettings”, “WifiAuto_On”, “EventCounter”, null
- [0192] When the user changes VPN automation to OFF, information sent is:
- [0193] “WifiSettings”, “VPNAuto_Off”, “EventCounter”, null
- [0194] When the user changes VPN automation to ON, information sent is:
- [0195] “WifiSettings”, “VPNAuto_On”, “EventCounter”, null
- [0196] When the user taps on Upgrade, information sent is:
- [0197] “WifiSettings”, “Upgrade”, “EventCounter”, null
- [0198] When the user taps on “. . .”, information sent is:
- [0199] “WifiSettings”, “More”, “EventCounter”, null
- [0200] When the user taps on “Forget Network”, information sent is:
- [0201] “WifiSettings”, “More_Forget”, “EventCounter”, null
- [0202] When the user taps on Advanced, information sent is:
- [0203] “WifiSettings”, “More_Advanced”, “EventCounter”, null
- [0204] When the user taps on Log, information sent is:
- [0205] “WifiSettings”, “More_Log”, “EventCounter”, null
- [0206] WiFi Settings Advanced
- [0207] FIG. 11 depicts an exemplary WiFi Settings Advanced screen shot. When the user opens this screen the name of the page he is looking at is sent to GA, in this case:
- [0208] WifiSettingsAdvanced
- [0209] User interactions with WiFi Settings Advanced:
- [0210] When the user unchecks Turn Wifi ON automatically, information sent is:
- [0211] “WifiSettings”, “Advanced_AutoOnDisabled”, “EventCounter”, null
- [0212] When the user checks Turn Wifi ON automatically, information sent is:
- [0213] “WifiSettings”, “Advanced_AutoOnEnabled”, “EventCounter”, null

- [0214] When the user unchecks Turn Wifi OFF automatically, information sent is:
- [0215] “WifiSettings”, “Advanced_AutoOffDisabled”, “EventCounter”, null
- [0216] When the user checks Turn Wifi OFF automatically, information sent is:
- [0217] “WifiSettings”, “Advanced_AutoOffEnabled”, “EventCounter”, null
- [0218] When the user taps on “Clear ID”, information sent is:
- [0219] “WifiSettings”, “Advanced_ClearID”, “EventCounter”, null
- [0220] Upgrade
- [0221] FIG. 12 depicts an exemplary WiFi Settings Advanced pair of screen shots. When the user opens this screen the name of the page he is looking at is sent to GA, in this case:
- [0222] Upgrade
- [0223] User interactions with Upgrade:
- [0224] When the user taps “Secure Me”, the information sent is:
- [0225] “Upgrade”, “Payment_Display”, “EventCounter”, null
- [0226] When the user complete payment process, the information sent is:
- [0227] “Upgrade”, “Payment_Ok”, “EventCounter”, null
- [0228] When the user taps “Cancel Subscription”, the information sent is:
- [0229] “Upgrade”, “Cancel”, “EventCounter”, null
- [0230] When the user taps on one secured hotspot, the information sent is:
- [0231] “Upgrade”, “SecuredHotspot_WifiSettings”, “EventCounter”, null
- [0232] Side Menu
- [0233] FIG. 13 depicts an exemplary Side Menu screen shot. The side menu may be accessed, as noted above, by a user tapping on the icon at the top left of the home screen, for example. When the user opens this screen the name of the page he is looking at is sent to GA, in this case:
- [0234] Side
- [0235] User interactions with Side Menus:
- [0236] When the user taps “Go Pro” (seen at top right of FIG. 13), the information sent is:
- [0237] “Side”, “Button_Upgrade”, “EventCounter”, null
- [0238] When the user change Wifi Assistant automation to OFF, information sent is:
- [0239] “Side”, “WifiAssistant_Off”, “EventCounter”, null
- [0240] When the user change Wifi Assistant automation to ON, information sent is:
- [0241] “Side”, “WifiAssistant_On”, “EventCounter”, null
- [0242] When the user taps “Share”, the information sent is:
- [0243] “Side”, “Button_Share”, “EventCounter”, null (also a screen event is sent as “Share” screen)
- [0244] When the user taps “More Info”, the information sent is:
- [0245] “Side”, “Button_About”, “EventCounter”, null
- [0246] When the user taps “News”, the information sent is:
- [0247] “Side”, “Button_News”, “EventCounter”, null
- [0248] When the user taps “Rate us”, the information sent is:
- [0249] “Side”, “Button_RateUs”, “EventCounter”, null
- [0250] About
- [0251] FIG. 14 depicts an exemplary About screen shot. A user can, for example, navigate to this screen from the side menu screen shown in FIG. 13. When the user opens the About screen the name of the page he is looking at is sent to GA, in this case:
- [0252] About
- [0253] When the user taps on these options a Screen event is sent
- [0254] Wifi interactions from Android System Settings
- [0255] Wifi interactions outside Wifi Assistant
- [0256] When a user turns off Wifi from Android settings Wifi Assistant is paused, and the following sent:
- [0257] “System”, “Assistant_Pause”, “EventCounter”, null
- [0258] When a user turns on Wifi from Android settings we also turn on Wifi in Wifi Assistant and sent:
- [0259] “System”, “Assistant_Resume”, “EventCounter”, null
- [0260] This event is also sent every time WiFi Assistant is resumed from the paused state. In exemplary embodiments of the present invention, this needs to be filtered in the analytics backend whenever resume was from a button pressed from the UI.
- [0261] As can be seen in the detailed application flow presented above, for an exemplary application such as “AVG WiFi Assistant”, by capturing the screens visited by a user, and the interactions of a user at each screen visited, systems and methods implementing exemplary embodiments of the present invention can use this behavior to operate on a database of many users, and predict churn or loyalty as to that application. If the method is applied to numerous applications of a single genre, say gaming applications, or social media applications, where correlations can be made between types of screens visited (e.g., all smartphone applications have an opening screen, a home screen and a user preferences set of screens), or, for example, to newer versions of existing programs with some changes, it is possible to predict churn or loyalty on a user interacting with a new application using a relatively small, or even no, training set and the data and predictions from all similar applications. Such a method can, for example, be an improvement over simply using the overall percentage of a small training set for the new app, as described above.
- [0262] Incorporation of Both User Characteristics and Behavioral Data in Models
- [0263] In the exemplary model generation process described above, the discriminating patterns are behavioral data, i.e., sequences of screens visited and events engaged in at such screens. It is most often the case that the differences in behavior as regards an app are discriminating as to propensity to churn the app. However, this is not categoric. Sometimes user characteristics are more predictive, or user characteristics in combination with various behavioral interactive sequences more predictive, of propensity to churn. It is thus noted that two customers with the same behavioral

sequences as regards an app may have quite different churn probabilities. There are some applications that are more friendly or geared to one customer demographic than another. For example, dating applications are more desired by women in their 30s than are fantasy football gambling applications. Similarly, bodybuilding applications are more inviting to younger males. Thus, in such demographic specific apps, it is often a combination of the user characteristics and a short behavioral sequence that can best discriminate as to churn propensity. Thus, any optimal clustering of customer characteristics and behavioral data may be useful in various exemplary embodiments of the present invention, and all such clusters, and the resultant discriminating patterns for churn or loyalty are contemplated, and within the scope of, the present invention.

V. NON-LIMITING SOFTWARE AND HARDWARE EXAMPLES

[0264] Exemplary Mobile Device and System

[0265] FIG. 16 shows a high-level block diagram of a mobile device 1601. It will be further appreciated that the device shown in FIG. 16 is illustrative and that variations and modifications are possible. Mobile device 1601 can include a controller 1602, a wireless module 1604, a location module 1606, churn prediction module 1608, a computer-readable medium (CRM) 1610, a display module 1612, and an input module 1614. Mobile device 1601 can include additional modules. In some embodiments, mobile device 1601 can be a sufficient size, dimension, and weight to enable the device to be easily moved by a user. For example, mobile device 1601 can be pocket size.

[0266] Controller 1602, which can be implemented as one or more integrated circuits, can control and manage the overall operation of mobile device 1601. For example, controller 1602 can perform various tasks, such as retrieving various assets that can be stored in CRM 1610, accessing the functionalities of various modules (e.g., interacting with other Bluetooth® enabled devices via a Bluetooth® module), executing various software programs (e.g., operating systems and applications) residing on CRM 1610, and so on. In some embodiments, controller 1602 can include one or more processors (e.g., microprocessors or microcontrollers) configured to execute machine-readable instructions. For example, controller 1602 can include a single chip applications processor. Controller 1602 can further be connected to CRM 1610 in any suitable manner.

[0267] Wireless module 1604 can include any suitable wireless communication technology. For example, wireless module 1604 could include a Bluetooth® module, a radio frequency (RF) module, a WiFi module, and/or the like. The Bluetooth® module can include any suitable combinations of hardware for performing wireless communications with other Bluetooth®-enabled devices and allows an RF signal to be exchanged between controller 1602 and other Bluetooth®-enabled devices. In some embodiments, a Bluetooth® module can perform such wireless communications according to Bluetooth® Basic Rate/Enhanced Data Rate (BR/EDR) and/or Bluetooth® Low Energy (LE) standards. The Bluetooth® protocol, in general, enables point-to-point wireless communications between multiple devices over short distances (e.g., 30 meters). Bluetooth® has gained widespread popularity since its introduction and is currently used in a range of different devices. In order to allow Bluetooth® to be used in a greater variety of applications, a

low energy variant of the technology was introduced in the Bluetooth® Core Specification, Version 4.0. Bluetooth® Low Energy (LE), in general, enables devices to wirelessly communicate while drawing low amounts of power. Devices using Bluetooth® LE can often operate for more than a year without requiring their batteries to be recharged.

[0268] For example, a Bluetooth® module can include suitable hardware for performing device discovery, connection establishment, and communication based on only Bluetooth® LE (e.g., single mode operation). As another example, a Bluetooth® module can include suitable hardware for device discovery, connection establishment, and communication based on both Bluetooth® BR/EDR and Bluetooth® LE (e.g., dual mode operation). As still another example, a Bluetooth® module can include suitable hardware for device discovery, connection establishment, and communication based only on Bluetooth® BR/EDR.

[0269] An RF module can include any suitable combinations of hardware for performing wireless communications with wireless voice and/or data networks. For example, an RF module can include an RF transceiver that enables a user of mobile device 1601 to place telephone calls over a wireless voice network.

[0270] A WiFi module can include any suitable combinations of hardware for performing WiFi-based communications with other WiFi-enabled devices. For example, a WiFi module may be compatible with IEEE 802.11a, IEEE 802.11b, IEEE 802.11g and/or IEEE 802.11n.

[0271] Location module 1606 can include any suitable location technology using one or more wireless signals to determine a current location. In some embodiments, location module 1606 includes a global positioning system (GPS) module. In some embodiments, location module 1606 includes one or more of the following: WiFi location module, cellular location module, crowd-sourced WiFi location module, time of flight calculations (ToF) location module, and the like.

[0272] Churn prediction module 1608 can include code that, when executed, predicts, based on a user's interaction with a given app also stored and operable on the mobile device, a probability that the user will churn the app, or be loyal to it. For example, using the methods described above, churn prediction module 1608 can send a prediction to a back end server operated by the app's publisher, for example. The app publisher can then, as described above, message the user in various attempts to persuade he or she to take actions which lessen the likelihood that he or she will churn. Moreover, the churn prediction module 1608 can continually download updated collective user data as well as algorithmic updates to fine tune its predictive models, and can, similarly, also perform device-side collection and aggregation of app usage data and transmit that to a back-end server.

[0273] CRM 1610 can be implemented, e.g., using disk, flash memory, random access memory (RAM), hybrid types of memory, optical disc drives or any other storage medium that can store program code and/or data. CRM 1610 can store software programs that are executable by controller 1602, including operating systems, applications, and related program code (e.g., code for churn prediction module 1608).

[0274] Software programs (also referred to as software or apps herein) can include any program executable by controller 1602. In some embodiments, certain software programs can be installed on mobile device 1601 by its manu-

facturer, while other software programs can be installed by a user. Examples of software programs can include operating systems, navigation or other maps applications, locator applications, productivity applications, video game applications, personal information management applications, applications for playing media assets and/or navigating a media asset database, applications for controlling a telephone interface to place and/or receive calls, and so on. Although not specifically shown, one or more application modules (or set of instructions) may be provided for launching and executing one or more applications, e.g., various software components stored in medium 1610 to perform various functions for mobile device 1601.

[0275] Display module 1612 can be implemented using any suitable display technology, including a CRT display, an LCD display (e.g., touch screen), a plasma display, a direct-projection or rear-projection DLP, a microdisplay, and/or the like. In various embodiments, display module 1612 can be used to visually display user interfaces, images, and/or the like.

[0276] Input module 1614 can be implemented as a touch screen (e.g., LCD-based touch screen), a voice command system, a keyboard, a computer mouse, a trackball, a wireless remote, a button, and/or the like. Input module 1614 can allow a user to provide inputs to invoke the functionality of controller 1602. In some embodiments, input module 1614 and display module 1612 can be combined or integrated. For example, mobile device 1601 can include an LCD-based touch screen that displays images and also captures user input. Illustratively, a user can tap his or her finger on a region of the touch screen's surface that displays an icon. The touch screen can capture the tap and, in response, start a software program associated with the icon. Upon starting the software program, a graphical user interface for the application can be displayed on the touch screen for presentation to the user.

[0277] Various exemplary embodiments of the invention as described above can be implemented as one or more program products, software applications and the like, for use with a computer system, e.g., a smartphone or other mobile user device. The terms program, software application, and the like, as used herein, are defined as a sequence of instructions designed for execution on a computer system or data processor. A program, computer program, or software application may include a subroutine, a function, a procedure, an object method, an object implementation, an executable application, an applet, a servlet, a source code, an object code, a shared library/dynamic load library and/or other sequence of instructions designed for execution on a computer system.

[0278] The program(s) of the program product or software may define functions of the embodiments (including the methods described herein) and can be contained on a variety of computer readable media. Illustrative computer readable media include, but are not limited to: (i) information permanently stored on non-writable storage medium (e.g., read-only memory devices within a computer such as CD-ROM disk readable by a CD-ROM drive); (ii) alterable information stored on writable storage medium (e.g., floppy disks within a diskette drive or hard-disk drive); or (iii) information conveyed to a computer by a communications medium, such as through a computer or telephone network, including wireless communications. The latter embodiment specifically includes information downloaded from the

Internet and other networks. Such computer readable media, when carrying computer-readable instructions that direct the functions of the present invention, represent embodiments of the present invention.

[0279] In general, the routines executed to implement the embodiments of the present invention, whether implemented as part of an operating system or a specific application, component, program, module, object or sequence of instructions may be referred to herein as a "program." The computer program typically is comprised of a multitude of instructions that will be translated by the native computer into a machine-readable format and hence executable instructions. Also, programs are comprised of variables and data structures that either reside locally to the program or are found in memory or on storage devices. In addition, various programs described herein may be identified based upon the application for which they are implemented in a specific embodiment of the invention. However, it should be appreciated that any particular program nomenclature that follows is used merely for convenience, and thus the invention should not be limited to use solely in any specific application identified and/or implied by such nomenclature.

[0280] It is also clear that given the typically endless number of manners in which computer programs may be organized into routines, procedures, methods, modules, objects, and the like, as well as the various manners in which program functionality may be allocated among various software layers that are resident within a typical computer (e.g., operating systems, libraries, API's, applications, applets, etc.) It should be appreciated that the invention is not limited to the specific organization and allocation or program functionality described herein.

[0281] The present invention may be realized in hardware, software, or a combination of hardware and software. A system according to a preferred embodiment of the present invention can be realized in a centralized fashion in one computer system, or in a distributed fashion where different elements are spread across several interconnected computer systems, including cloud connected computing systems and devices. Any kind of computer system—or other apparatus adapted for carrying out the methods described herein—is suited, and preferably the present invention is implemented in a smartphone, tablet or other personal electronic device. A typical combination of hardware and software could be a general purpose computer system with a computer program that, when being loaded and executed, controls the computer system such that it carries out the methods described herein. On the user device side, for example, a typical combination of hardware and software could be receiver provided with one or more data processors with a computer program that, when being loaded and executed, controls the data processors such that they carry out the methods described herein.

[0282] Each computer system may include, inter alia, one or more computers and at least a signal bearing medium allowing a computer to read data, instructions, messages or message packets, and other signal bearing information from the signal bearing medium. The signal bearing medium may include non-volatile memory, such as ROM, Flash memory, Disk drive memory, CD-ROM, and other permanent storage. Additionally, a computer medium may include, for example, volatile storage such as RAM, buffers, cache memory, and network circuits. Furthermore, the signal bearing medium may comprise signal bearing information in a transitory state medium such as a network link and/or a network interface,

including a wired network or a wireless network, that allow a computer to read such signal bearing information.

[0283] Although specific embodiments of the invention have been disclosed, those having ordinary skill in the art will understand that changes can be made to the specific embodiments without departing from the spirit and scope of the invention. The scope of the invention is not to be restricted, therefore, to the specific embodiments. The above-presented description and figures are intended by way of example only and are not intended to limit the present invention in any way except as set forth in the following claims. For example, while this disclosure speaks in terms of predicting churn or loyalty probabilities for an application on a mobile telephone, as noted above, its techniques and systems are applicable to any type of application, on any type of user device. It is particularly noted that persons skilled in the art can readily combine the various technical aspects of the various elements of the various exemplary embodiments that have been described above in numerous other ways, all of which are considered to be within the scope of the invention.

What is claimed:

1. A processor-implemented method for predicting user churn, the method comprising:

collecting, using one or more data processors:

user data corresponding to a user of an application program running on a user device, the user data including user basic attribute information, and user interaction data associated with the application program, including at least one of (i) which user interface screens were visited, (ii) in which sequence, and (iii) which events were engaged in at each screen;

determining, using the data processors, a similar user group for the user based on the user data;

determining, using the data processors, one or more discriminating patterns from the user interaction data; selecting at least one of said discriminating patterns according to a defined set of rules,

calculating a probability that the user will churn or be loyal to the application program; and

at least one of: storing the probability on the user device, and transmitting the probability for the user to a server.

2. The method of claim 1, wherein the similar user group is assigned based on either (i) clustering done on one of a training set or (ii) updated clustering performed by a back-end server.

3. The method of claim 1, wherein the similar user group is assigned based on initial clustering done on a training set, as periodically updated using all then available user data,

4. The method of claim 1, wherein the discriminating patterns comprise one of: a sequence of user interface screens visited by the user, or a sequence of user interface screens visited by the user and the actions taken at each user interface screen.

5. The method of claim 1, wherein the selected discriminating pattern is chosen based on length, being the longest pattern.

6. The method of claim 1, wherein of multiple discriminating patterns the longest is chosen, and wherein if there exist multiple discriminating patterns of equal length, the one with the highest churn probability is chosen.

7. The method of claim 1, wherein a probability that a user will churn is calculated after each user interaction with a user interface screen.

8. The method of claim 7, wherein in response to a probability above a defined level indicating churn, messages are sent to the user to direct the user to visit one or more specific user interface screens to diminish the probability of churning.

9. A non-transitory computer-readable medium including one or more sequences of instructions that, when executed by one or more processors, cause:

collecting:

user data corresponding to a user of an application program running on a user device, the user data including user basic attribute information, and user interaction data associated with the application program, including at least one of:

(i) which user interface screens were visited,

(ii) in which sequence, and

(iii) which events were engaged in at each screen;

determining of a similar user group for the user based on the user data;

determining of one or more discriminating patterns from the user interaction data;

selecting at least one of said discriminating patterns according to a defined set of rules; and

calculating a probability that the user will churn or be loyal to the application program.

10. A computer system comprising:

one or more processors; and

a memory accessible to the one or more processors, the memory storing instructions executable by the one or more processors to:

collect:

(i) user data corresponding to a user of an application program running on a user device, the user data including user basic attribute information, and

(ii) user interaction data associated with the application program, determine a similar user group for the user based on the user data;

determine one or more discriminating patterns from the user interaction data;

select at least one of said discriminating patterns according to a defined set of rules,

calculate a probability that the user will churn or be loyal to the application program; and

at least one of:

store the probability on the user device, and transmit the probability for the user to a server.

11. The computer system of claim 10, wherein the similar user group is assigned based on either (i) clustering done on one of a training set or (ii) updated clustering performed by a back-end server.

12. The computer system of claim 10, wherein the similar user group is assigned based on initial clustering done on a training set, as periodically updated using all then available user data,

13. The computer system of claim 10, wherein the discriminating patterns comprise one of: a sequence of user interface screens visited by the user, or a sequence of user interface screens visited by the user and the actions taken at each user interface screen.

14. The computer system of claim 10, wherein the selected discriminating pattern is chosen based on length.

15. The computer system of claim **10**, wherein of multiple discriminating patterns the longest is chosen, and wherein if there exist multiple discriminating patterns of equal length, the one with the highest churn probability is chosen.

16. The computer system of claim **10**, wherein a probability that a user will churn is calculated after each user interaction with a user interface screen.

17. The computer system of claim **16**, wherein in response to a churn probability above a defined level, messages are sent to the user to direct the user to visit one or more specific user interface screens to diminish the probability of churning.

18. The computer system of claim **10**, further comprising at least one of: storing the probability on the user device, and transmitting the probability for the user to a server.

19. The computer system of claim **10**, wherein said calculating a probability is performed on a server, and in response to a churn probability above a defined level, messages are sent to a user device to direct the user to visit one or more specific user interface screens to diminish the probability of churning.

20. The method of claim **1**, wherein said calculating a probability is performed on a user device, and user data is uploaded from the user device to proprietary or cloud servers.

21. The method of claim **20**, wherein a more detailed churn analysis, using up to the minute collective data for the given app, is performed on the servers.

* * * * *