Albayram and Khan *Hum. Cent. Comput. Inf. Sci.* (2016) 6:16
DOI 10.1186/s13673-016-0072-3

Human-centric Computing and Information Sciences

RESEARCH Open Access

Evaluating smartphone-based dynamic security questions for fallback authentication: a field study

Yusuf Albayram* and Mohammad Maifi Hasan Khan

*Correspondence:
yusuf.albayram@uconn.edu
Department of Computer Science and Engineering, University of Connecticut, Storrs, CT, USA

Abstract

To address the limitations of static challenge question based fallback authentication mechanisms (e.g., easy predictability), recently, smartphone based autobiographical authentication mechanisms have been explored where challenge questions are not predetermined and are instead generated dynamically based on users' day-to-day activities captured by smartphones. However, as answering different types and styles of questions is likely to require different amounts of cognitive effort and affect users' performance, a thorough study is required to investigate the effect of type and style of challenge questions and answer selection mechanisms on users' recall performance and usability of such systems. Towards that, this paper explores seven different types of challenge questions where different types of questions are generated based on users' smartphone usage data. For evaluation, we conducted a field study for a period of 30 days with 24 participants who were recruited in pairs to simulate different kinds of adversaries (e.g., close friends, significant others). Our findings suggest that the question types do have a significant effect on user performance. Furthermore, to address the variations in users' accuracy across multiple sessions and question types, we investigate and present a Bayesian classifier based authentication algorithm that can authenticate legitimate users with high accuracy by leveraging individual response patterns.

Keywords: Authentication, Usability, Security, Fallback authentication, Autobiographical authentication, Security questions, Smartphones, Android

Background

While password based schemes are prevalent forms of authentication, as the number and complexity of passwords continue to grow due to the increasing number of online accounts per user and complicated password creation policies, users are finding it increasingly difficult to manage and remember passwords for different accounts, and have to reset passwords frequently [1, 2]. To address this, prior efforts investigated various forms of fallback authentication mechanisms to facilitate resetting of passwords. Among these, pre-selected challenge questions (i.e., personal knowledge questions) are often used as a fallback authentication mechanism to facilitate resetting/recovery of passwords [3]. However, this widely used approach of leveraging static challenge questions as fallback authentication mechanism has several limitations. For instance, recent studies revealed that personal knowledge based questions are often susceptible to

various weaknesses such as easy predictability, inapplicability, and poor recall rate [3–8]. Furthermore, static security questions are becoming weaker due to improved information retrieval techniques and increases in online content [8], where an attacker can obtain the answers to many of the static challenge questions by mining online sources (e.g., social networking sites, public records or even a simple Google search).

To address the limitations of static security questions, recently, smartphone based autobiographical authentication mechanisms have been explored where challenge questions are not predetermined and generated dynamically based on users' day-to-day activities captured by smartphones. Specifically, the constantly changing information about the daily behavior (e.g., phone calls, location traces) of a person is used to generate one-time authentication questions (e.g., Who did you call around 4:30 pm today?, What apps did you use in the last 24 h?).

Such dynamic security questions have several potential advantages over static challenge questions. For instance, as dynamic security questions are generated on the fly, users do not have to configure these questions in advance. Also, while certain static security questions are often not applicable for some users (e.g., What is the name of your first pet?), or may be easily found by looking at someone's Facebook or LinkedIn profile (e.g., "What year did you graduate from college?"), dynamic security questions are likely to be harder to guess by mining online sources due to the randomness of a person's smartphone usage behavior and day-to-day activities. While a limited number of prior efforts looked into the possibility of using dynamic security questions for fallback authentication, they only looked at a limited number of question types [9–11], making it hard to judge the strengths and weaknesses of this approach considering different kinds of users, question types, and answer selection schemes (e.g., open-ended vs. multiple choice).

To complement these prior efforts and examine the effect of different types of security questions on different categories of users' recall/guessability performance (e.g., legitimate user, adversarial users), this paper explores the design space of dynamic security questions that are generated using users' day-to-day activities captured by smartphones. Specifically, we evaluate the recall rate and guessability rate for different question types for different kinds of adversarial users (e.g., naive vs. knowledgeable) through a field study over a period of 30 days with 24 participants who were recruited in pairs to simulate different kinds of adversaries (e.g., close friends, significant others). Over the course of the study, each participant was periodically presented with three sets of challenge questions. The first set of question was generated based on participant's own data. The second set of question was generated based on participant's pair's (e.g., close friend or couple) data. In this case, the role of strong adversary is played by the pair of each participant. Finally, the third set of question was generated based on a randomly selected participant's data. In this case, participants played the role of naive adversary.

Over the course of the study, we collected a total of 7672 responses for seven different question types. Subsequently, we analyzed the response accuracy for different types of questions to gain insight regarding how they are perceived by users (e.g., easy to remember, hard to guess). Our findings suggest that the type of questions (e.g., communication event, location) and the style of questions (e.g., multiple choice, open ended) have a significant impact on both legitimate and adversarial users' response accuracies. Finally, as performance and response patterns of individual users can vary significantly for different

question types, we investigate and present a Bayesian classifier based authentication algorithm that can authenticate legitimate users with high accuracy by leveraging individual response patterns while reducing the success rate of adversaries.

To summarize, this paper makes the following key contributions:

1. First, we present the first study that compares the usability of seven different dynamic security question types, namely, call, SMS, location, application usage, music, physical activity, and battery charging events, which are generated based on users' smartphone usage behavior and day-to-day activities captured by smartphones.
2. Second, we conduct a field study that investigates the strengths and weaknesses of the presented schemes against different kinds of adversarial users (e.g., naive vs. knowledgeable) for different categories of questions.
3. Finally, we discuss the usability aspects of such systems based on qualitative feedback collected using an exit survey.

The rest of the paper is organized as follows. "Related work" section presents a summary of prior work. "Methods" section explains the study design and describes the algorithms for question generation and score calculations. "Evaluation" section presents the results from our experimentation with real users. Insights gained in this work along with limitations of our study and future directions of our work are discussed in "Discussion and limitations of the study" section. Finally, "Conclusion" section concludes the paper.

Related work

Prior efforts investigated various forms of fallback authentication mechanisms to facilitate resetting of passwords or to provide an extra layer of security for authentication. The most widely known fallback authentication mechanism is based on challenge questions (a.k.a, security questions) where a service (e.g., website) requires a user to answer personal verification questions such as "what is the name of your first pet?". However, the use of such pre-agreed personal authentication questions has been widely criticized and considered to be a weak form of authentication [3–5, 7, 8, 12] due to various vulnerabilities. For example, Zviran and Haga [13] conducted a study in which they found that participants were able to remember 78 % of their answers to a set of personal security questions, and those who are close to the participants (e.g., spouses, close friends) were able to guess the answers correctly 33 % of the time. In line with this, Podd et al. in 1996 [14] conducted a similar study and reported a similar recall rate (80 %) for legitimate users and higher guessability from attackers (39.5 %). Both studies reported that participants forget 20–22 % of their answers within three months. Schechter et al. [7] conducted a user study where they evaluated the challenge questions used by four large webmail providers and they pointed out that 20 % of the participants forget their own answers within 6 months. 17 % of their answers was guessed correctly by acquaintances with whom participants were unwilling to share their webmail passwords. 13 % of the answers could be guessed within five attempts by guessing the most popular answers of other participants. More recently, Bonneau et al. [4] analyzed a real-world data set on security questions of millions of Google account holders and they found that a significant fraction of users (40 %) were unable to recall their answers when needed. Also, they

pointed out that questions that are more memorable such as city of birth and father's middle name are also the easiest to find from the public records or social networking sites, while potentially safest questions suffer from low recall rate (e.g., frequent flyer number only has a 9 % recall rate). In another study, Rabkin [8] identified that security questions are getting weaker due to improved information retrieval techniques and increases in online content. By mining online sources (e.g., social networking sites, public records and search engines), an attacker can obtain the details about one's personal information to answer many of the challenge questions commonly used for fallback authentication. For example, the answer to the question "What year did you graduate from college?" may be found from one's Facebook profile or LinkedIn profile. Rabkin [8] found that 16 % of the questions had answers publicly available in online social networking sites. Moreover, as many of the challenge questions are often used across different websites, the consequences of compromising a single account can be overwhelming.

To address the limitations of static challenge question based schemes, several new approaches have been investigated where challenge questions are not predetermined and are dynamically generated on the fly based on user's recent activities such as online browsing history [15, 16], Facebook activity [17], electronic personal history (e.g., personal calendar data) [18, 19], email history [20], and users' recent location history [21, 22]. Among works that looked into leveraging smartphone usage data and behavior data, Das et al. [9] proposed to generate authentication questions from users' day-to-day activities captured by their smartphones. However, the authors did not evaluate this scheme against various attackers in real-life (e.g., close friends). Hang et al. [10] presented the design of dynamic security questions generated based on smartphone usage data (e.g., call and SMS histories, app usage). Along this line, we investigated location-based dynamic challenge question generation schemes where different types of questions are generated based on users' locations tracked by smartphones and presented to users for fallback authentication [22].

While several works have investigated the possibility of using dynamic security questions for authentication, only a handful of them looked into the aspect of memorability. Among these, recently, Gupta et al. [23] investigated the memorability of smartphone usage behavior (e.g., calls, texts, emails) and attempted to leverage that to authenticate users. One of the main limitations of this work is that the challenge questions are generated based on a user's routine (e.g., who do you call the most?) rather than day-to-day activities which are more dynamic and is the focus of our work. In another work, Hang et al. [24] proposed a fallback authentication system where users are asked to remember the arrangements of icons on home screen. However, accurately remembering the icon arrangement on home screen can be difficult for users, since some apps can change the arrangement of home screen icons (e.g., after uninstalling/installing an app), which hinders the usability of such systems. More recently, Hang et al. [11] presented a fallback authentication mechanism where users are asked to recognize apps that are installed or not installed on their smartphones.

While these and our own exploratory studies [25, 21, 22] present interesting results, no comprehensive study has been done so far that looked into the strengths and weaknesses of such systems considering different kinds of users, question types, and answer selection schemes (e.g., open-ended vs. multiple choice). Towards that, our work presented

in this paper complements these prior efforts that investigated the challenge of designing dynamic security questions using users' smartphone usage behavior data for fallback authentication, and extends prior efforts in several ways as follows. First, to the best of our knowledge, we are the first to explore several new data types (e.g., Music play-list history, Physical activity, Battery charging events) and answer types (e.g., multiple choice vs. open ended, time selection) for generating dynamic security questions for fallback authentication. Second, we conducted a real life study in which we evaluate the strengths and weaknesses of dynamic security questions for different categories of questions and user types (e.g., legitimate, naive adversary, strong adversary). Finally, we attempt to understand users' perceptions regarding dynamic security questions through an interview style exit survey. The details of our work are presented in the following sections.

Methods

In this section, we first describe the smartphone application that was developed to collect and analyze autobiographical data. We then present the algorithms for question generation and score calculations. Finally, we present the design of the study. The details are below.

Data collection

We developed an Android application that supports devices running Android 2.3 or higher to collect autobiographical user data, and then analyzes the collected data to generate challenge questions. Table 1 lists the details regarding the data that are collected in our study.

The communication data (call, SMS) was obtained from the recent communication history. The app usage data is estimated based on how long an app is in the foreground while the smartphone screen is on. Music playlist history was obtained through the broadcast event of music player (e.g., default music player, Spotify). Battery charging events were collected when the smartphone was connected to a power source (e.g., AC, USB). Physical activity of users was obtained using the Android activity recognition API [26] which provides an easy way to recognize five different types of user activities (e.g., riding in a vehicle, riding a bicycle, walking, running, and no movement). In order to obtain the location information with minimal energy overhead, we utilize the latest Google Fused Location API [27] along with Android's activity recognition API [26]. Specifically, the application leverages the Android's activity recognition API to decide

Table 1 Details of the collected data

Data	Details of collected data
Call	Type (outgoing, incoming), duration, name of the person, time
SMS	Type (sent, received), receiver/sender name, length of SMS message, time
App usage	App name, package name, duration, time
Music	Track name, artist, album, duration, time
Physical activity	Type (walking, running, in vehicle, on bicycle), confidence level, duration, time
Battery charging	Type of power connection (AC, USB), duration, time
Location	Latitude, longitude, duration, time, accuracy (i.e., the expected error bound)

whether to track location or not. For example, when a user is not moving, the app does not track location at all. On the other hand, if a user is detected to be walking, biking, or riding in a vehicle, the app starts logging location data.

Once the data items are collected, the question generation component generates challenge questions as follows.

Autobiographical question generation

Using the aforementioned data types, the application generates nine different types of questions as listed in Table 2. Please note that we only use data from the last 24 h to generate questions. This is based on the prior work which showed that users can recall a good number of many episodic memories that are one day old compared to events that are older [28]. Details about each type of questions are below.

Questions generated based on communications activity

Communication questions are generated based on a user's recent communication history (e.g., SMS history that includes both sent and received messages, and call history that includes both incoming and outgoing phone calls). This category of questions asks a user to recall the name of the person he/she called or SMS messaged, or the name of the person who called him/her or SMS messaged him/her at a certain time. Examples of communication questions are shown in Fig. 1a, b. For this type of question, a user is asked to enter the answer (i.e., person's name) into a textbox. To enhance the usability, we utilize the "auto-complete" feature which suggests possible entries as a user types in the textbox. This is especially helpful as "auto-complete" feature reduces potential errors due to possible misspellings and speeds up the process, especially when entering a long text (e.g., long name and last name of a person). As an attacker may take advantage of this feature (i.e., "auto-complete"), especially if a user has a very limited number of contacts (i.e., limited answer space), the application inserts random names which are derived from an online fake name generator [29]. Inserting fake names in addition to users' contact list names increases the answer space significantly, making it harder to guess the correct answer, which is also reflected in the exit survey.

Table 2 List of question types generated by the application

Question type	Question	Retrieval type
Call	Who called you on <time> ? Who did you call on <time> ?	Recall
SMS	Who SMS messaged you on <time> ? Who did you SMS message on <time> ?	Recall
Location	Where were you on <time> ?	Recall
Application	What are the applications you used in the last 24 h?	Recognize
Music	What are the music you listened to in the last 24 h?	Recognize
Activity	What activities did you perform in the last 24 h and when?	Recognize and recall
Battery	When did you charge your phone in the last 24 h and how was it charged?	Recognize and recall

Fig. 1 Screenshots showing different types of questions. **a** Call, **b** SMS, **c** App, **d** Music **e** Activity, **f** Battery and **g** Location

Questions generated based on application usage data

Application usage questions are generated based on a user's recent app usage history. This type of question is a recognition question in which a user is presented with 15 options and asked to identify the applications that he/she used within the last 24 hours. Options are presented along with icons of the applications to facilitate recall. An example app question is shown in Fig. 1c. The presented 15 options contain multiple correct answers along with multiple incorrect answers (i.e., distractors). In particular, a random number of correct answers are sampled from the user's recent app usage history by using the algorithm, which is explained in "Algorithm for generating challenge questions" section. Also, when sampling the correct answers, we avoided selecting apps that are preinstalled by operating system such as home screen and launcher apps (e.g., "TouchWiz home launcher"). This was done for two reasons. First, these apps are commonly pre-installed by manufacturers of devices without the knowledge of users [11], thus these apps are often not seen as being apps by users [30]. Second, these apps (e.g., home launcher) may be too obvious for an adversary to guess. A random number of plausible incorrect answers are compiled based on the user's past app usage history that are not

used recently. Other distractors are derived from the top downloaded apps in Google Play Store [31]. In this paper, we present 15 options to make the answer sufficiently hard to guess [21].

Questions generated based on music data

Questions about music are generated based on a user's recent music playlist history which contains the log of the music a user has played. Similar to app questions, this type of question is a recognition question in which a user is presented with 15 options and is asked to identify the music that he/she played within the last 24 h. The presented list may contain multiple correct answers along with multiple incorrect answers (i.e., distractors). A random number of correct answers are sampled from the user's recent music playlist history by using the algorithm presented in "Algorithm for generating challenge questions" section. A random number of plausible incorrect answers are compiled based on the user's past music playlist history that have not been played recently. Other distractors are derived from the top tracks in Spotify by using Spotify Web API [32]. Furthermore, this API provides a wide range of genre categories along with search criteria, which allows us to select distractors based on track genres, artists, and albums. Options are presented with album cover art along with 30 s audio previews of tracks using the Spotify Web API [32]. When a play button is tapped, 30 s of audio from the corresponding track starts playing (see Fig. 1d). The use of icons and audio previews provides a more appealing and simpler interface for the user. Moreover, it facilitates recall for users by providing auditory and visual cues (e.g., album thumbnails). An example of a music question is shown in Fig. 1d.

Questions generated based on physical activity log

Questions about physical activities are generated based on the log that records the physical activities of a user. In this question type, a user is asked to: (1) identify all the physical activities that he/she performed longer than a specified threshold (e.g., 3 min) within the last 24 h and (2) select the time window for the selected activity by using the time slider. An example physical activity question can be seen in Fig. 1e. Note that this type of question can have multiple correct answers (i.e., multiple different physical activities can be performed during a day). For instance, if a user was in a vehicle twice during a day, the user can pick a time for any of these activities (i.e., in vehicle activity).

Questions generated based on battery charging events

Questions that are generated based on recent battery charging events ask a user to identify: (1) the time when the device was plugged into a power source (i.e., charger) within the last 24 h and (2) the mode of charging (e.g., AC charger, USB) (see Fig. 1f). As in the physical activity question, this type of question can have multiple correct answers as well (i.e., a device may be charged multiple times during a day). In case of multiple correct answers, a user needs to pick only one of them.

Questions generated based on location information

Location questions are generated based on a user's recent location traces tracked by the application. The collected location data is composed of a sequence of coordinates with

latitude, longitude and the relevant temporal information (i.e., time stamp). To avoid considering each geographical coordinate as a unique physical location, in our work, we use a clustering algorithm that groups geographical coordinates based on their distance in order to infer user's locations. However, as a user may visit new places over time and we do not know a priori the total number of places a user may visit, we chose to use a density-based clustering approach that can incrementally adapts the number of clusters (i.e., the number of distinct physical locations). Specifically, we employ the DBSCAN (density-based spatial clustering of applications with noise) algorithm [33] that is based on the notion of density reachability. Briefly, the DBSCAN algorithm requires two parameters: ϵ distance threshold (e.g., 75 m) and min_{pts} minimum number of points within a cluster. The algorithm starts by assigning a random point in a cluster and expands it with neighborhoods of at least min_{pts} points that are within a distance ϵ from it. In our work, to calculate the distance between two geographical coordinates, we use Haversine distance [34], though the algorithm can work with any distance function. Based on the distance of examined coordinates and ϵ distance threshold, the DBSCAN algorithm either creates new clusters or expands/updates the existing clusters. As new coordinates arrive, new coordinates are first examined to determine whether they can be assigned to any of the existing clusters. If not, the new coordinates are given as input to the DBSCAN algorithm to regenerate the clusters including the new locations. The output of this algorithm is a set of clusters that is used to generate questions. Please note that this algorithm only runs whenever the application needs to generate location questions for a user. For location-based questions, a user is presented an interactive map and is asked to select the location that he/she had visited during a certain time window of a specific day (see Fig. 1g). The interactive map was implemented leveraging the Google Maps Android API [35] where the initial zoom level was set to one to make the most of the world visible. The rationale behind this choice is to avoid influencing users to select locations from a certain geographic area, which may reduce the overall security of the system [36]. In order to select a location on the map, the minimum required zoom level is set to 16, which gives reasonable details and higher security since an adversary has to guess a location at a finer resolution. A user needs to long press on the map to pin his/her location and set a marker at the selected location (e.g., like the one in Fig. 1g). Instead of zooming in/out manually, user may also use a search box implemented using Google Place Autocomplete feature to zoom-in on the right area/location very quickly.

Algorithm for generating challenge questions

As there can be hundreds of events per day (e.g., phone call events, SMS messages), it is nontrivial to pick the specific instance of an event that may be used to generate the question. Ideally, the system should pick an event that is easy for a legitimate user to recall but hard for an adversary to guess.

To address this challenge, we develop an algorithm that gives preference to rare events compared to more predictable events. Intuitively, if a person rarely receives a phone call from person X, it is more likely that he/she will remember that event. This intuition is also supported by prior work in psychology. For instance, Kristo et al. [37] reported that events that occur less frequently are remembered better compared to events that occur frequently.

To implement the algorithm, we represent a user's history H as a sequence of events (e.g., phone call). In H, each event is represented as a triplet of the form $e_i = (a_i, d_i, t_i)$, where a_i represents an activity (e.g., making a phone call), d_i is the duration of this activity and t_i is the time-stamp of the event. Assuming that n activities were recorded for a user in a given time frame, the history for that time frame will be represented as a time ordered sequence of triplets, and will be denoted as $H = \{e_1, ..., e_n\}$. Subsequently, we convert the history H into a time window-activity matrix as shown in Table 3 by splitting each day into a set of m time windows $W = \{w_0, ..., w_m\}$ of fixed size (e.g., 1 h). Next, each event is assigned to a specific time windows W_i based on the event's time-stamp t_i.

Once events are assigned in specific time windows, the system computes an "interestingness" weight for each event based on statistical measure of randomness, and attempts to pick events for generating questions by giving preference to more infrequent events in a user's schedule. To identify the infrequent events for a given *Time window-location matrix* (e.g., as shown in Table 3), the algorithm analyzes daily and weekly activity patterns of a user and calculates the weight for an event as follows.

1. First, the algorithm calculates $P(e_i)$ which denotes the probability of an event e_i. For example, probability of calling John in the last 30 days based on call log data.
2. Next, the algorithm calculates $P(e_i|w_m)$ which denotes the probability of event e_i for a specific time interval w_m. For example, the probability of calling John between 10:00 am and 11:00 am in the last 30 days. This probability is calculated to identify daily patterns.
3. Next, the algorithm calculates $P(e_i|w_m, dow_k)$ where dow_k denotes the "day of the week" from the set $DOW = \{dow_1, ..., dow_7\}$ where $dow_1 = Monday, \ldots, dow_7 = Sunday$. $P(e_i|w_m, dow_k)$ denotes the probability of an event e_i for a time interval w_m on day dow_k of the week. For example, the probability of calling John between 10:00 am and 11:00 am on Mondays in the last 30 days. This probability is calculated to identify weekly patterns.
4. Finally, to give priority to long lasting events which are more likely to be remembered by a user easily, the algorithm calculates T_e^i which denotes the sum of the duration of event e_i in the history H and subsequently, multiply with d_i (duration of the event).

Table 3 Time window-event matrix of a user's phone call log history

Window\day	Nov 14	Nov 15	...	Dec 27
00 : 00 — 00 : 59	–	–	...	{*Receivedcallfrom — Jeff*, 55*s*}
01 : 00 — 01 : 59	–	–	...	–
⋮	⋮	⋮	⋮	⋮
14 : 00 — 14 : 59	{*Called — Alice*, 55 *s*}, {*Called — Bob*, 32 *s*}	{*Called — Bob*, 89 *s*}	...	{*Called — Bob*, 17 *s*}
15 : 00 — 15 : 59	{*Called — John*, 300 *s*}	{*Receivedcallfrom — Jeff*, 42 *s*}	...	–
16 : 00 — 16 : 59	{*Called — John*, 14 *s*}	{*Called — Bob*, 20 *s*}	...	{*Called — Bob*, 89 *s*}
17 : 00 — 17 : 59	{*Called — Bob*, 27 *s*}	–	...	–
⋮	⋮	⋮	⋮	⋮
23 : 00 — 23 : 59	–	{*Receivedcall — Bob*, 14 *s*}	...	{*Called — Mike*, 14 *s*}

Numbers right next to a person's name indicates the duration of the phone calls in seconds

For example, multiply a recent phone call duration made to John with the sum of the duration of phone calls that made to John in the last 30 days based on call log data. The main intuition behind this multiplication is that we want to give priority to the latest events that lasted longer compared to other events of the same type.

5. Based on the above probabilities, we compute the *weight* of an event as follows:

$$Weight = \frac{P(e_i)P(e_i|w_m)P(e_i|w_m, dow_k)}{T_e^i \times d_i}$$

Once weight for individual events are calculated, the algorithm sorts all events based on *weight* and picks according to that order whenever the system needs to generate challenge questions for a particular data type. Please note that the lower the weight, the higher the chance of that event to be selected by the algorithm.

Due to the above scheme, higher weight questions that are relatively easy to guess because of "regularity" are filtered out and the preference is given to more infrequent events which are more likely to be harder to guess but easier to recall by legitimate users. Please note that the above scheme can be applied for any data types such as call log, SMS log, and location log. However, necessary changes may need to be made based on data types. For instance, for SMS log, there is no duration for SMS messages, and thus duration needs to be ignored or may be replaced with the length of SMS messages. For location log, to avoid considering each geographical coordinate as a unique physical location, geographical coordinates are clustered first.

User score calculation

To be able to authenticate users based on their response accuracy, we need to calculate the score of a user for a given session where multiple different questions may be asked for authentication. However, as users may make different kinds of mistakes for different question types while answering the challenge questions, we need to calculate the score differently for different question types. Table 4 lists the question formats and the mechanisms that were used to calculate scores for different question types. We describe how scores are calculated for different types of questions below.

Score calculation for communication questions

In the case of communication questions (i.e., Call and SMS), a user can either pick the answer from a list of suggestions that are populated using an "auto-complete" functionality, or a user may type in his/her answer instead of selecting from the list of names suggested by the "auto-complete" feature. However, while typing, a user may make spelling mistakes. For example, a common first name "Adrianna" may be spelled as "Adrianne" or "Adrienne". Thus, instead of scoring the answer based on an exact match (where the correct answer and the user's answer are matched 100 %), in the current implementation, there is an error tolerance to accommodate typing errors (e.g., 85 % similarity score between two strings). Specifically, if the Jaro-Winkler distance [38] between the entered text and the correct answer has a similarity score greater than 85 %, the answer is considered to be correct. Otherwise, the score is set to 0. Please note that the Jaro-Winkler distance metric is considered to be well-suited for comparing short strings such as person names [38].

Table 4 List of question types along with their corresponding question format, score calculation and existence of distractors

Question type	Question format	Score calculation	Distractors
Call	Open-ended	Jaro-Winkler [38] see "Score calculation for communication questions" section	✗
SMS	Open-ended	Jaro-Winkler [38] see "Score calculation for communication questions" section	✗
Location	Open-ended	Haversine [34] see "Score calculation for location questions" section	✗
Application	Multiple-choice	Eq. 1 see "Score calculation for app usage and music questions" section	✓
Music	Multiple-choice	Eq. 1 see "Score calculation for app usage and music questions" section	✓
Activity	Multiple-choice and time selection	Eq. 2 see "Score calculation for activity and battery charging questions" section	✗
Battery	Multiple-choice and time selection	Eq. 2 see "Score calculation for activity and battery charging questions" section	✗

Score calculation for location questions

For location questions, as users may not place the marker on exactly the same location coordinates estimated and identified by the system, there is an error tolerance (e.g., 75 m great circle distance) in our system, which is calculated based on the Haversine distance formula [34] between the selected coordinates and the estimated location. Specifically, if the distance between the selected geographical location and the estimated location is greater than 75 m, the answer is considered to be incorrect and the score is set to 0. Please note that this threshold (75 m) has been identified based on prior work [22], where it was shown to be useful to distinguish between legitimate and adversarial users.

Score calculation for app usage and music questions

In the case of questions about application usage and questions about the music played, a user is presented with 15 options where he/she can choose multiple answers from the given set of options. We develop a simple mechanism based on the methods in [39] to calculate a partial score for a particular question. Specifically, a user receives points if he/she correctly selects an answer, but if the user picks an incorrect answer, he/she is penalized (i.e., receives negative points). Please note that due to the penalization scheme, a user may receive a negative score, which may happen if he/she selects more incorrect options than correct options.

For a given application or a music question q, a score is calculated as follows:

$$Score^{q}_{app} = \frac{1}{n^{q}} \left(n^{q}_{ac} - \frac{n^{q}_{aw}}{sp} \right) \tag{1}$$

where

- n^q: the number of options that are correct for a question q.
- n^q_{ac}: number of selected options for which the answer is correct for question q.
- n^q_{aw}: number of selected options for which the answer is wrong for question q.
- sp: severity of penalty is a parameter that controls the points deducted/subtracted for an incorrect answer.

In Eq. 1, sp is a parameter that controls how many points are deducted for an incorrect answer. The lower the sp, the higher the penalty for an incorrect answer. In our implementation, we chose $sp = 1$, which indicates that the penalty for an incorrect answer is equal to the points for a correct answer. Hence, positive score indicates more correct answers than incorrect answers, while negative score implies the opposite. The rationale behind this score calculation scheme is to: (1) prevent statistical guessing attacks where someone can simply select all possible answers to get the question correct (i.e., random guesser), and (2) allow users to get partial scores when they make a small number of mistakes (e.g., one correct app is ignored). For example, consider an app usage question that has 15 options, and 4 out of 15 options are correct. In this case, if a user selects 5 options where 4 of them is correct, 1 of them is incorrect and we choose $sp = 1$, the score for this question will be $\frac{1}{4}(4 - \frac{1}{1}) = 0.75$. If a user selects 4 options where 1 of them is correct, 3 of them is incorrect and $sp = 1$, the score will be $\frac{1}{4}(1 - \frac{3}{1}) = -0.5$. Note that in the first example, the user is slightly penalized ($score = 0.75$), losing one of his/her correct answers though he/she selected all four correct answers. On the other hand, in the second example, the user is severely penalized and gets negative score ($score = -0.5$) due to selecting more incorrect answers than correct answers.

Please note that we use the same formula (i.e., Eq. 1) to compute scores for music questions.

Score calculation for activity and battery charging questions

In the case of physical activity and battery charging questions, users are asked to select the correct events and the time for the selected events. As the answer consists of two parts, we calculate the score by giving different weights for each part of the answer (e.g., event type and the time of the event) as shown in Eqs. 2 and 3. Intuitively, as guessing the correct time of an event is more likely to be harder than guessing the type of the event, we give higher weight to the time component of the answer compared to the event component where a user selects the type of the event. Specifically, we set $w_t > w_o$ in Eq. 3 so that precisely answering the time component of an answer contributes more to the score compared to the event type of the answer. Furthermore, as a user may not be able to specify the exact time of the event, we use a maximum allowed time t_{max} and a minimum allowed time t_{min} threshold to award or penalize points depending on how "close" the answer is to the correct answer. Specifically, we consider three scenarios as shown in Eq. 4: (1) if the difference between a user's answer and the true time is beyond t_{max} (e.g., 2 h), the user receives negative points (i.e., -1 point) for the time component of the answer, (2) if the difference between a user's answer and the true time is less than t_{min} (e.g., 1 h), the answer is considered to be correct and the user receives full point (i.e.,

1 point), and (3) if the difference between a user's answer and the true time is less than t_{max} and greater than t_{min}, the user receives partial points depending on how close the answer is to the true time (i.e., correct answer). The closer the selected time to the true time, the higher the awarded points will be.

Note that w_t and w_o can be adjusted to give higher weights to different parts of the answer, and t_{min} and t_{max} can be adjusted to allow different fidelity for the time window. For instance, if $w_t > w_o$, more weight is given to the time component of the answer. In our evaluation, for activity and battery charging questions, we set $w_o = 0.2$ and $w_t = 0.80$ where $\sum w_o + w_t = 1$, $t_{min} = 60$ min and $t_{max} = 120$ min, meaning that if the difference between a user's answer and the true time is smaller than t_{min} (e.g., 1 h), the user's answer is considered to be correct, and if the differences is within the range of t_{min} and t_{max}, the user receives partial points.

$$Score^{q}_{activity} = \frac{1}{n^q}\left(w^{q}_{n_{ac}} - \frac{n^{q}_{aw}}{sp}\right) \tag{2}$$

where

$$w^{q}_{n_{ac}} = \sum_{i=1}^{n^{q}_{ac}} w_o + w_t * f(t_{diff}) \tag{3}$$

where

$$f(t_{diff}) = \begin{cases} 1 & \text{if } t_{diff} \leq t_{min} \\ min\left(\frac{t_{max} - t_{diff}}{t_{max}}\right) & \text{if } t_{min} < t_{diff} \leq t_{max} \\ -1 & \text{otherwise} \end{cases} \tag{4}$$

where

$$t_{diff} = min\Big(\mid t_{i,correct} - t_{i,selected} \mid \Big)$$

where

- n^{q}_{ac}: number of selected options for which the answer is correct for a question q.
- w_o: weight for the event type component of the answer
- w_t: weight for the time component of the answer
- t_{diff}: time differences between the selected answer and the correct time(s)
- t_{min}: minimum allowed time difference
- t_{max}: maximum allowed time difference
- $t_{i,correct}$: time of the correct answer for a question
- $t_{i,selected}$: selected time for a question

Model-based authentication

As different users often differ in terms of mental ability to recall past events, they are expected to perform differently in answering different questions. Hence, in our work,

we leverage users' historical performance and response patterns in addition to accuracy score to authenticate users. Specifically, we evaluate a Bayesian classifier based authentication algorithm and compare the performance against a simple threshold based scheme. They are presented below in increasing order of complexity.

Threshold based scheme

As a single question may not be enough for reliably authenticating a user, we assume that multiple questions may be asked in a single session. Hence, in this scheme, we calculate the score of a user by calculating the average accuracy over multiple challenge questions. As a user may answer only a subset of questions correctly, ideally, it should be possible for someone to get access even with less than perfect score. To accommodate imperfect score, in our system, we calculate the authentication rate of legitimate users using a global threshold based scheme, where a user is identified as a legitimate user if his/her score is greater than some predefined threshold δ. We vary the value of δ from 100 to 0 % in our study.

While the threshold based scheme is easy to understand and apply, as each individual user is different and may perform differently, the threshold based scheme performs poorly (see "Classification accuracy of threshold based scheme" section). To address this, we attempt to account for this variations by building Bayesian classifier based models for each user based on individual response patterns, which is described next.

Bayesian based classifier for authentication

To account for variations in individual response patterns and accuracy, instead of relying solely on user's accuracy score and expecting the user to answer all questions correctly, the system learns a user's response pattern, and subsequently leverages the response patterns along with accuracy score to authenticate the user. For example, a user who usually answers call questions correctly but SMS questions incorrectly would be more likely to answer call questions correctly and SMS questions incorrectly in future attempts (i.e., repeat a similar pattern). Using this scheme, even if an adversary somehow can observe and learn a user's daily activities and answers all the questions correctly, the adversary will require to closely imitate the response errors and behavior of a legitimate user to gain access to the system (e.g., time takes to answer).

To this end, in our work, we use a Bayesian based classifier which is inspired based on prior work [9] that has shown that Bayesian classifier is appropriate to distinguish between legitimate and adversarial users with high accuracy, which is a similar context to ours (i.e., they evaluated performance of this classifier for similar attack scenarios). We also considered several advantages of Bayesian classifiers such as ease of implementation, speed in training and classification, and the fact that they can be used for real time prediction [40–42].

In this model, to predict whether a given response comes from a legitimate user (i.e., u) or an adversary (i.e., u') based on k response features ($f_1 ... f_k$), we create separate models for each question type for each user. For example, for user 1, we have seven different models for seven different question types.

Let's assume that for each question type Q_i, we have n responses $(r_1, ..., r_n)$ which are obtained from n different sessions for a user. Each such response r_i can be represented

by the response features ($f_1..f_k$). Hence, Naïve Bayes Classifier for this case can be written as follows for each question type Q_i and for each response r_i:

$$P(u|f_1..f_k) = \frac{P(f_1..f_k|u)P(u)}{P(f_1..f_k)}, \tag{5}$$

where $P(u|f_1..f_k)$ (i.e., *posterior*(u)) is the probability of being a legitimate user based on the response features. $P(u)$ is the prior probability distribution of the legitimate user. We assume that the chance of being a legitimate user is 50 % (i.e., equal probability), so $P(u) = P(u') = 0.5$. $P(f_1..f_k|u)$ represents the joint probability of responses given a user is legitimate. Since f_i's are independent (based on our assumption), $P(f_1..f_k|u)$ can be rewritten as the product of the component probabilities as follows:

$$P(f_1..f_k|u) = p(f_1|u)p(f_2|u)...p(f_k|u),$$

The denominator $P(f_1..f_k)$ of Eq. 5 represents the joint probability of responses' features. This can be expressed as follows:

$$P(f_1..f_k) = P(u)P(f_1..f_k|u) + P(u')P(f_1..f_k|u'),$$

where $P(f_1..f_k|u')$ denotes the probability of being an adversary (i.e., non-user) based on the response features.

To evaluate the classification accuracy of this scheme for different users and different attack scenarios, we build two models. First, we build one model for each legitimate user using the historical data of the user. Second, we build one model that represents the community of attackers. Since it would be quite unrealistic to assume that the attacker data is known by the system in advance, we use this simple alternative for more realistic classifier evaluation of different attack scenarios. More specifically, we train the attacker's model using the community of attackers without assuming known data from the attacker for a user (see Fig. 2). As shown in Fig. 2, we split each of legitimate and

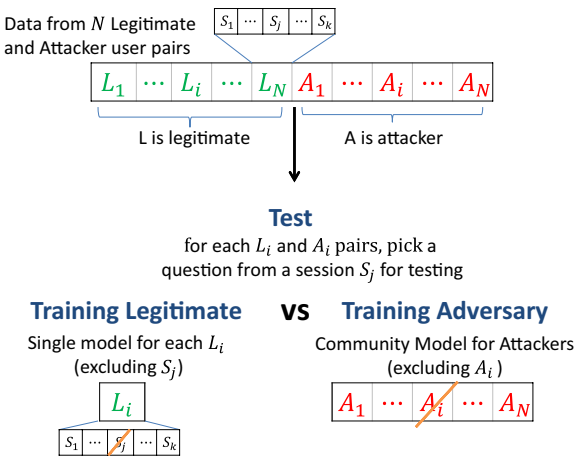

Fig. 2 Training and test scheme for evaluation of the Bayes classifier. For training, we build 2 models: (1) a model for each legitimate user, (2) a community model for attackers excluding the corresponding legitimate user's attacker data. Since it is unrealistic to assume that attackers' data is known in advance by the system, we use this scheme for more realistic evaluation of the classifier

attacker user pair's response data into k folds where k denotes the number of sessions. Subsequently, we use data from $(k-1)$ sessions to train the model and use the remaining session for testing. We repeat the process k times where each time we use a different session for testing. To test the system, we try three different attack scenarios (against strong adversaries, against naive adversaries, and against the community of both strong and naive adversaries) as follows.

In the first case, we assume the existence of only strong adversaries in the system (i.e., all attackers are strong adversaries). In this case, the classifier is trained on data from the legitimate user and the community of all strong adversaries excluding the corresponding legitimate user's strong adversary as shown in Fig. 2. Once the model is constructed, the test dataset from the legitimate user and the corresponding legitimate user's strong adversary is used to assess the classification accuracy of the system against unknown strong adversary without assuming known data from an individual attacker. In the second case, we assume the existence of only naive adversaries in the system (i.e., all attackers are naive adversaries who are trying to compromise the system without any knowledge regarding the daily routines of the targeted user). In this case, the classifier is trained on data from the legitimate user and community of all naive adversaries excluding the corresponding legitimate user's naive adversary. As before, the test dataset consists of legitimate user's data and his/her naive adversary data. Finally, in the third case, we do not distinguish between naive and strong adversaries and the classifier is trained using data from the community of strong and naive adversaries in the system. Again, we exclude the corresponding legitimate user's strong and naive adversaries' data from the training dataset.

Study design

To evaluate the security and usability of the dynamic security question mechanism, we recruited 24 participants from a college campus through the university email list server.

To simulate strong adversaries, we recruited participants in pairs (e.g., close friends, significant others). The social relationships between the pairs of participants are as follows. Four participants brought their significant other, and eight participants brought their close friends. We also asked participants to rate how well they know each other on a Likert-scale of 1 (Very little) to 5 (Pretty well). A majority of the participants reported that they knew the partner pretty well (median = 5, mode = 5 and mean = 4.6).

Over the course of the experiments, each participant was presented with three sets of questions multiple times each week. The first set of question was generated based on participant's own data. For example, a participant would receive a phone call question in the following format: "who did you call at 11:25 am on Wednesday?". The second set of question was generated based on participant's pair's (e.g., close friend) data. In this case, the role of a strong adversary is played by the pair of each participant. For example, the participant would receive a phone call question about his/her partner in the following format: "who did your partner call at 4:20 pm on Friday?". The third set of question was generated based on a randomly selected participant's data whose identity was not revealed to the participant who answered the questions. In this case, participants played the role of a naive adversary. For example, the participant would receive a phone call question about a stranger in the following format: "who did a stranger call at 2:51 pm on

Monday?". In all cases, participants were not given any feedback regarding his/her performance throughout the study to avoid biasing them. Moreover, while participants were attempting to answer the challenge questions generated based on their close friend's data or a random user's data (i.e., playing the role of an adversary), the exact same questions and the exact same possible answer options (if a question has multiple answer choices such as app question—Fig. 1c) that were presented to the legitimate user are presented to the adversarial users. Each participant was compensated with a $25 Amazon gift card for two weeks of participation. The study was approved by the University's Institutional Review Board (IRB).

Evaluation

During a period of 30 days, we collected a total number of 7672 valid question-answer responses from 24 participants (24 legitimate users, 24 strong adversarial users and 24 naive adversarial users). Table 5 lists the breakdown of the number of responses collected for seven different question types and three user types (i.e., legitimate, strong and naive adversarial users). Out of 7672 responses, 2865 responses were from legitimate users, 2553 responses were from strong adversarial users, and 2254 responses were from naive adversarial users. One of the participants withdrew from the study after two weeks of participation. All participants (10 female, 14 male) were undergraduate students from a broad range of degree programs (e.g., psychology, material science and engineering). The age of participants ranged from 18 to 23 years with an average age of 19.33 years (Median = 19 and SD = 1.28). The key findings are discussed below.

Descriptive statistics for collected data

We collected smartphone usage data including call logs, SMS logs, location logs, application usage logs, music play list history, physical activity logs, and battery charging logs from 24 participants for over a month.

Figure 3a shows the statistics for phone call data including outgoing and incoming phone calls. The histogram plot appears to be right-skewed as most of the participants (~80 %) made 1–4 phone calls on average per day during the study period (i.e., 1 month).

Figure 3b shows the statistics for SMS data including the number of sent and received text messages. The histogram for the SMS data appears to be right-skewed as most of the participants (~90 %) sent and received 50–100 SMS messages on average per day.

Table 5 Number of question-answer responses collected for each question and user type

Question type	Number of response collected		
	Legitimate	Strong	Naive
Call	288	267	235
SMS	523	488	406
Location	480	452	388
Activity	347	289	271
Battery	416	346	313
App	437	349	308
Music	374	362	333
Total	2865	2553	2254

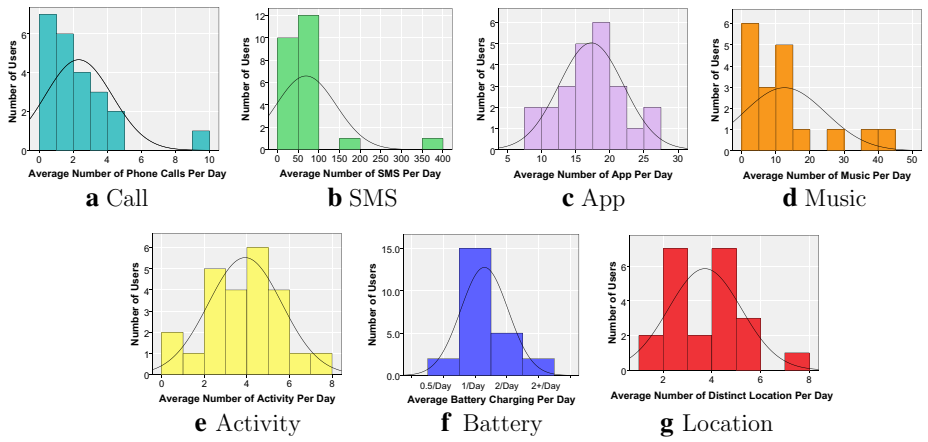

Fig. 3 *Histograms* of average number of phone calls (**a**), SMS (**b**), application used (**c**), songs listened to (**d**), activity performed (**e**), battery charging events (**f**) and distinct locations (**g**) per day across all users

Although one of the participants received and sent 150–200 SMS on average per day and another participant received and sent 350–400 SMS messages on average per day. It is worth noting that the number of SMS messages sent/received is relatively higher than the number phone calls made/received. This indicates that our participants, who were college students, preferred text-messaging service over phone calls. This may be because texting is quicker and more convenient in many situations (e.g., loud environment) and texting rates are generally cheaper than calling rates.

Figure 3c shows the average number of applications used by a participant per day during the study period. The histogram appears to be centered. All of the participants used on average at least five different apps per day and most of them used 15–20 different apps on average per day. Figure 3d shows that most of the participants listened to 1–15 songs on average per day while two of the participants listened to as many as around 45 songs on average per day. Here, we only considered music that is listened to more than 30 s. Figure 3e shows the statistics for physical activity data. We found that most of the participants (∼90 %) did not perform either running or bicycling, while many (∼ 80 %) walked and rode a vehicle between 2–6 times on average per day. Here, we only considered activities that had duration of more than 3 min. Figure 3f shows the statistics for battery charging events. We found that most of the participants (15) charged their phones on average once/day, 5 of the participants twice/day, 2 of them more than twice/ day and 2 of them every other day. Figure 3g shows the number of distinct locations visited per day by participants during the study period. The histogram shows that most of the participants visited 2–6 unique locations on average per day. Intuitively, as users send/receive a lot of SMS messages, we expected that SMS related questions will be harder to answer compared to questions that are based on other data types (e.g., phone call, application usage, music, physical activity, battery charging, and location). Furthermore, as a user needs to specify the time of the events in case of physical activity and battery charging related questions, we expected these types of open-ended questions to be relatively harder compared to multiple choice application usage and music questions where users only need to identify the correct answer(s) from a given list. We expected users' confidence to vary as well across question types. Our findings are below.

Accuracy scores

Table 6 shows average accuracy scores across different question types and user types. For legitimate users, the accuracy score of call, SMS, location, app, music, activity and battery related questions are 0.76, 0.46, 0.69, 0.55, 0.46, 0.42 and 0.53 respectively. For strong adversarial users, the accuracy score of call, SMS, location, app, music, activity and battery related questions are 0.13, 0.08, 0.29, −0.03, −0.71, 0.06 and −0.005 respectively. In terms of guesability, naive adversarial users performed much worse than strong adversarial users, their accuracy score varied between 0.038 and −1.782. The negative accuracy scores for adversarial users mainly stems from the penalization scheme where incorrect answers are penalized (i.e., gets negative score)—[recall that score calculations are explained in "User score calculation" section (Table 4)]. Please note that, due to the different score calculation schemes, score ranges are varied for different question types, and thus scores are not comparable. In order to evaluate how well each security question work and compare against different security question types, we used ROC curves, which is a standard way of comparing performance [43] (see "Accuracy of model-based authentication" section). Hence, although the accuracy scores given in Table 6 are not comparable, ROC curves shown in Figs. 4 and 5 are comparable across different question types and user types.

In summary, legitimate users performed significantly better compared to adversarial users in terms of answer correctness. As we expected, strong adversarial users performed better than naive adversarial users, since strong adversarial users have significant knowledge regarding legitimate users compared to naive adversarial user. We found that it was very difficult for a naive adversary to guess users' smartphone usage behavior and day-to-day activities.

Users' level of confidence

During the study, after answering a question, users (i.e., both legitimate and adversarial users) were asked to rate their level of confidence in their answer on a 5-point Likert scale where 1 means "Not confident at all" and 5 means "Very confident". To analyze the confidence ratings, which may passively indicate whether legitimate users find a particular question type easy to recall, or easy to guess in case of adversarial users, we used Kruskal–Wallis test which is a non-parametric alternative of ANOVA. Since the response data is ordinal, non-parametric Kruskal-Wallis tests are more appropriate in this context. The Kruskal-Wallis test demonstrated that there were significant differences in confidence

Table 6 Average accuracy scores for each question and user type

Question type	Accuracy score		
	Legitimate	Strong	Naive
Call	0.76	0.13	0.008
SMS	0.46	0.08	0.002
Location	0.69	0.29	0.038
App	0.55	−0.03	−0.549
Music	0.46	−0.71	−1.782
Activity	0.42	0.06	−0.240
Battery	0.53	−0.005	−0.157

ratings of users among the seven types of questions for three different user types, ($\chi^2(6) = 60.22$, $p < 0.01$) for legitimate users, ($\chi^2(6) = 27.23$, $p < 0.01$) for strong adversarial users, and ($\chi^2(6) = 19.52$, $p = 0.03$) for naive adversarial users. For demonstration purposes in Fig. 6, we provide the distributions (the median and interquartile range (IQR)) of confidence ratings among the seven types of questions for three different user types. The medians are indicated by the horizontal bold lines for each boxplot. Two whiskers (upper and lower) show greatest and least values respectively excluding outliers. Small circles shown below lower whiskers (e.g., the first subplot that shows legitimate user's confidence ratings) show outliers, which indicates that these observations are less than 1.5 times of lower quartile (i.e., 25 % of data is less than this value). Small circles shown above upper whiskers (e.g., the first subplot that shows naive adversarial users' confidence ratings) show outliers, which indicates that these observations are more than 1.5 times of upper quartile (i.e., 25 % of data is greater than this value). Also, extreme outliers are marked with a star (e.g., the subplot shows call questions confidence rating for naive adversarial users).

As shown in this figure, legitimate users were generally confident in their answers (medians are 5 for all question types except Music and SMS questions (median: 4)). For SMS questions, part of the reason may be because the participants sent/received a large number of SMS messages (mean: 69, median: 56), and they may text several people within a span of few minutes, making it difficult to recall reliably due to a large volume of the events. As listening to music can be a passive activity where users may do it while doing other things and/or often listen to whatever plays automatically instead of choosing a specific song each time, users may find it harder to remember and answer music questions, which is reflected in their low confidence rating.

When it comes to adversarial users, strong adversarial users appeared to be generally neutral (median: 3) in their confidence ratings. Confidence ratings of naive adversarial users were generally close to the lowest rating 1. Moreover, for comparison, we present the mean rank values (which were obtained from the Kruskal–Wallis test) and their rank orders among the seven question types for legitimate and adversarial users in Table 7. For each user type, we compare the mean rank values among the question types. The question type with the highest mean rank is considered to have higher confidence ratings for a user type. Please note that the mean rank values are compared by order of magnitude for only comparing question types for a particular user type. For instance, application usage questions had the highest confidence rating for legitimate users. For strong adversaries, location questions had the highest confidence rating (i.e., strong adversarial users were more confident when guessing location questions compared to other six question types) followed by activity questions. For naive adversarial users, battery and location questions had the highest confidence ratings. Intuitively, these results make sense, since location and activity of legitimate users can be observed by their close friends (a close friend may know the routines of the target user). On the other hand, adversarial users were less confident when guessing call and SMS questions, as these data types are more personal and hard to guess. Moreover, naive adversarial users had higher confidence ratings for location questions, which may stem from the fact that they knew that the other participants were from the same campus/locality.

Time taken to answer questions

During the study, we kept track of the amount of time that was taken by a user to answer a question. Table 8 shows the descriptive statistics about the time taken for different types of users to answer different types of questions. We found that adversarial users (i.e., strong and naive adversarial users) generally took less time on average to answer the questions compared to legitimate users. Furthermore, we observed that time taken varied for question types. For example, the longest time was taken for location questions with mean 28.56 s (median: 23 s) for legitimate users. Part of the reason may be due to the fact that users had to zoom in at level 16 (at least) to select a location on Google Map. Also, text based questions (Call and SMS questions) seemed to take less time than multiple choice questions (Application and Music). The mean time for Call and SMS questions are 15.51 (median: 11) and 17.99 (median: 12), while for application and music questions are 18.70 (median: 14) and 20.55 (median: 13) s respectively. Moreover, we found the lowest mean time for battery questions with mean 14.15 (median: 9, SD: 21:44) s, which was very close to the mean time for text based questions (i.e., Call and SMS).

Accuracy of model-based authentication

As performance metric, ROC (Receiver operating characteristics) plots are commonly used for evaluating classification performance, in which TPR (true positives rate) on the Y-axis is plotted as a function of FPR (false positives rate) on the X-axis. A ROC plot shows the tradeoff between TPR and FPR for all possible thresholds (i.e., cut-off points) [44]. In our context, the true positive rate (TPR) corresponds to the success rate of legitimate users, while the false positive rate (FPR) denotes the success rate of adversaries. We also use the area under the ROC curve (AUC) to measure the accuracy of the test. Note that the performance of the test can be quantified with a single value by calculating the AUC [43] which is an important indicator of the classification performance. AUC $= 0.5$ represents a test performed at chance for binary classification (i.e., the model performs no better than a coin flip), while AUC $= 1$ means a perfect test where all legitimate users succeed and adversaries failed to enter the system. Hence, the larger the AUC value, the better the model/test. Please note that, while evaluating performance of both threshold and Bayesian classifier based model, we use AUC value for different attack scenarios, question types and vary the number of questions using the data collected from our field study.

In this work, we first present the performance of a simple threshold based scheme, and next compare that against the performance of a more sophisticated Bayesian classifier based model. The details are below.

Classification accuracy of threshold based scheme

We generate ROC curves for the threshold-based scheme to show how the number of questions would affect TPR and FPR in identifying users for two different attack scenarios. In particular, in the first scenario, we assume the existence of only strong adversaries in the system (i.e., all attackers are strong adversaries), while in the second scenario, we assume the existence of only naive adversaries in the system (i.e., all attackers are naive adversaries). The three curves in each plot in Figs. 4 and 5 are generated for different numbers of questions (n) ranging from 1 to 6 for these two attack scenarios. For brevity,

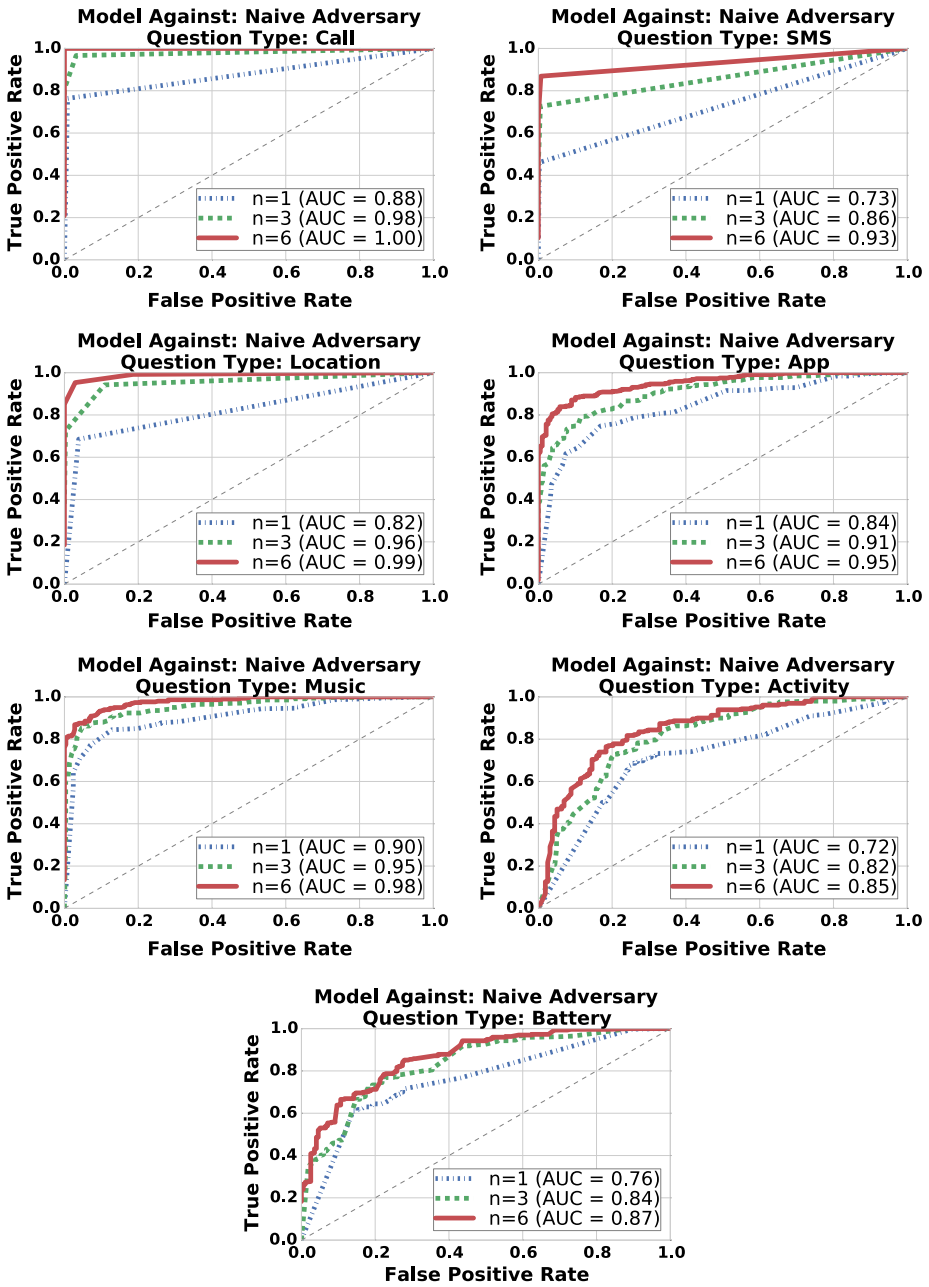

Fig. 4 Receiver operating characteristic (ROC) curves of the threshold based scheme for different types of questions when modeled against strong adversaries for $n = 1$, $n = 3$, and $n = 6$ questions respectively

we only show results for $n = 1$, $n = 3$, and $n = 6$ for a given question type and attack scenario. From these figures, it can be seen that, although the performance is better when modeled against naive adversaries in Fig. 5 compared to strong adversaries in Fig. 4, the performance of threshold based scheme against strong adversaries is not impressive. This motivates us to explore the Bayesian classifier based scheme which is explained next.

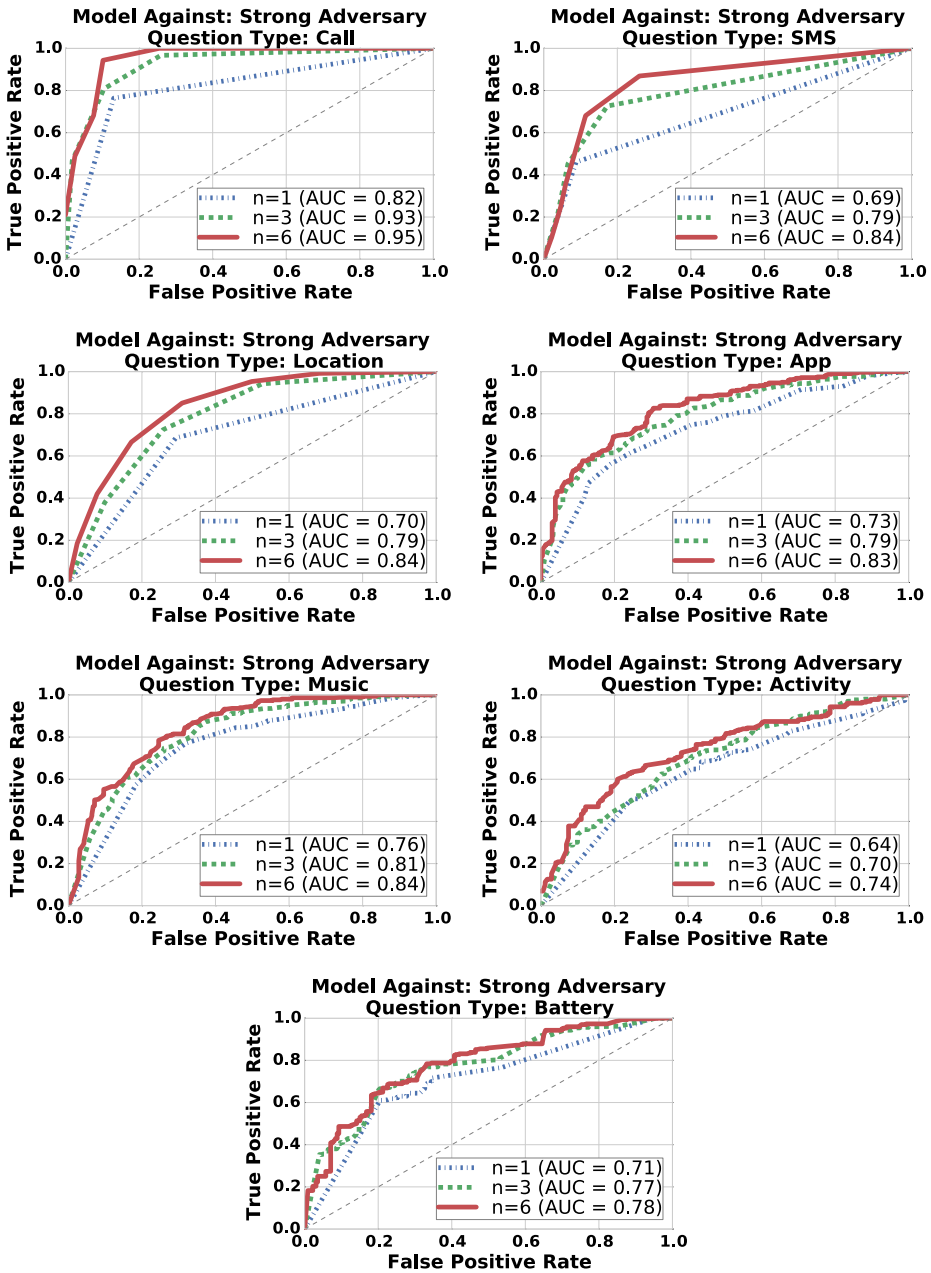

Fig. 5 Receiver operating characteristic (ROC) curves of the threshold based scheme for different types of questions when modeled against naive adversaries for $n = 1, n = 3$, and $n = 6$ questions respectively

Accuracy of Bayesian based classifier for authentication

We generated ROC curves (as shown in Figs. 7, 8 and 9) to capture the tradeoff between TPR and FPR while using different number of questions answered and different attack scenarios modeled. The three curves displayed in each plot in Figs. 7, 8 and 9 are generated for $n = 1$, $n = 3$, and $n = 6$ for the three attack scenarios, namely, against naive adversaries, against strong adversaries, and against both strong and naive adversaries respectively. In these ROC Figures, the AUC value of each test is shown as the

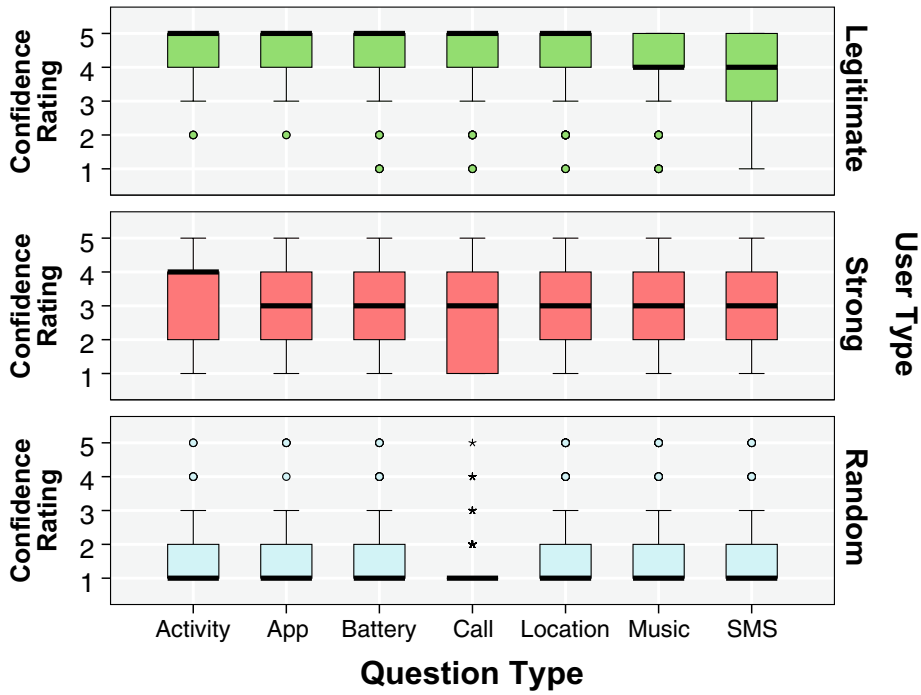

Fig. 6 Legitimate and adversarial users' confidence ratings for seven different question types. During the study, with every question answered, all users were asked to rate their level of confidence in their answer on a scale of 1 (Not confident at all) to 5 (Very Confident)

Table 7 The ordered "mean rank" of confidence ratings for seven different question types for three different user types

Order	Legitimate	Strong	Naive
1	App (888)	Location (857)	Battery (778)
2	Activity (880)	Activity (833)	Location (747)
3	Location (848)	App (830)	App (734)
4	Battery (805)	Music (802)	Activity (717)
5	Call (801)	Battery (793)	Music (702)
6	Music (776)	SMS (725)	SMS (672)
7	SMS (651)	Call (690)	Call (646)

The higher the mean rank, the more confident users were in their answers. The mean rank values are designated within parenthesis right next to the question types. Please note that the mean rank values are compared by order of magnitude for only question types for a particular user type

performance of the test. We observe that answering more question increases the AUC value regardless of the modeled adversary. In other words, the system becomes increasingly confident in identifying legitimate users as the number of questions answered within a category increases. For example, the AUC values are 0.88 for $n = 1$, 0.95 for $n = 3$ and 1.0 for $n = 6$ when modeled against strong adversary and the AUC values are 0.96 for $n = 1$, 0.99 for $n = 3$ and 1.0 for $n = 6$ when modeled against naive adversary for call question. The four performance measures are summarized in Table 9 where we used four performance measures TPR, FPR, Accuracy and F1 score to evaluate performance of the Bayesian classifier [45]. For brevity, we only show results for $n = 1$, $n = 3$, and $n = 6$

Table 8 Time taken for legitimate and adversarial users to answer different types of questions

	Legitimate			Strong			Naive		
	Mean	Median	SD	Mean	Median	SD	Mean	Median	SD
Activity	18.11	13	17.15	13.96	9	15.47	8.45	6	9.95
Battery	14.15	9	21.44	10.42	7	11.47	7.00	4	7.41
App	18.70	14	14.43	18.68	14	17.16	14.75	12	11.00
Music	20.55	13	28.88	25.79	17	31.77	13.73	9	21.96
Call	15.51	11	13.20	7.53	5	9.52	6.96	5	7.26
SMS	17.99	12	17.88	7.88	5	9.91	7.34	5	7.57
Location	28.56	23	19.05	18.79	15	12.98	14.94	11	11.05

The time unit is in seconds

for a given question type and attack scenario. From Table 9, it can be seen that regardless of the modeled adversary, legitimate users were able to obtain high accuracies after answering more questions. In particular, for all question types, as the value of n increases from 1 to 6, TPR, F1 score and accuracy rate increase for legitimate users. Also, we see that the classification performance varied greatly depending on the modeled adversary. For example, for $n = 3$ and when modeled against strong adversary, the average accuracy rates are 96.0 % for call questions, 94.1 % for SMS questions, 96.3 % for location questions, 97.1 % for app questions, 96.8 % for music questions, 98.0 % for activity questions and 95.9 % for battery questions. On the other hand, when modeled against strong adversary and $n = 3$, the average accuracy rates are slightly lower–93.1 % for call questions, 86.8 % for SMS questions, 85.8 % for location questions, 93.4 % for app questions, 85.9 % for music questions, 86.8 % for activity questions and 93.6 % for battery questions. Intuitively, since a strong adversary has significant knowledge regarding a user's schedule (e.g., girlfriend), strong adversarial users are more likely to gain access to the system by answering questions more accurately compared to naive adversarial users. Furthermore, among the question types, phone call, app usage, and battery charging questions generally perform better compared to SMS, location, activity, and music questions.

User's opinions regarding autobiographical authentication

To gain insight into the perceived usability and security of smartphone based autobiographical authentication systems, at the end of the study, the participants were asked to complete an exit survey for an additional $10 Amazon Gift card. We asked the participants to rate several statements on a five-point Likert-scale where 1 indicates strong disagreement and 5 indicates strong agreement with a given statement.

Table 10 summarizes the survey result. As the survey responses are ordinal data, we report median and mode rather than mean and standard deviation for each question response. Also, in case where multiple modes exist, the smallest value is shown in Table 10. Our key findings are below.

First, most of the participants found phone call, location, app, music, and battery charging based questions to be relatively easier to recall (mode 4 median 4, mode 5 median 4, mode 4 median 4, mode 5 median 4 and mode 5 median 4 respectively)

compared to SMS and Activity based questions (mode 4 median 3.5 and mode 3 median 3 respectively), as expected.

Second, when it comes to guessability, most of the participants disagreed that guessing the answers of their close friend's question were easy. Also, majority of the participants strongly agreed that a stranger will not be able to guess answers to their questions easily.

Finally, based on our exit survey, users were found to be generally positive about the idea of answering dynamic security questions for fallback authentication instead of static

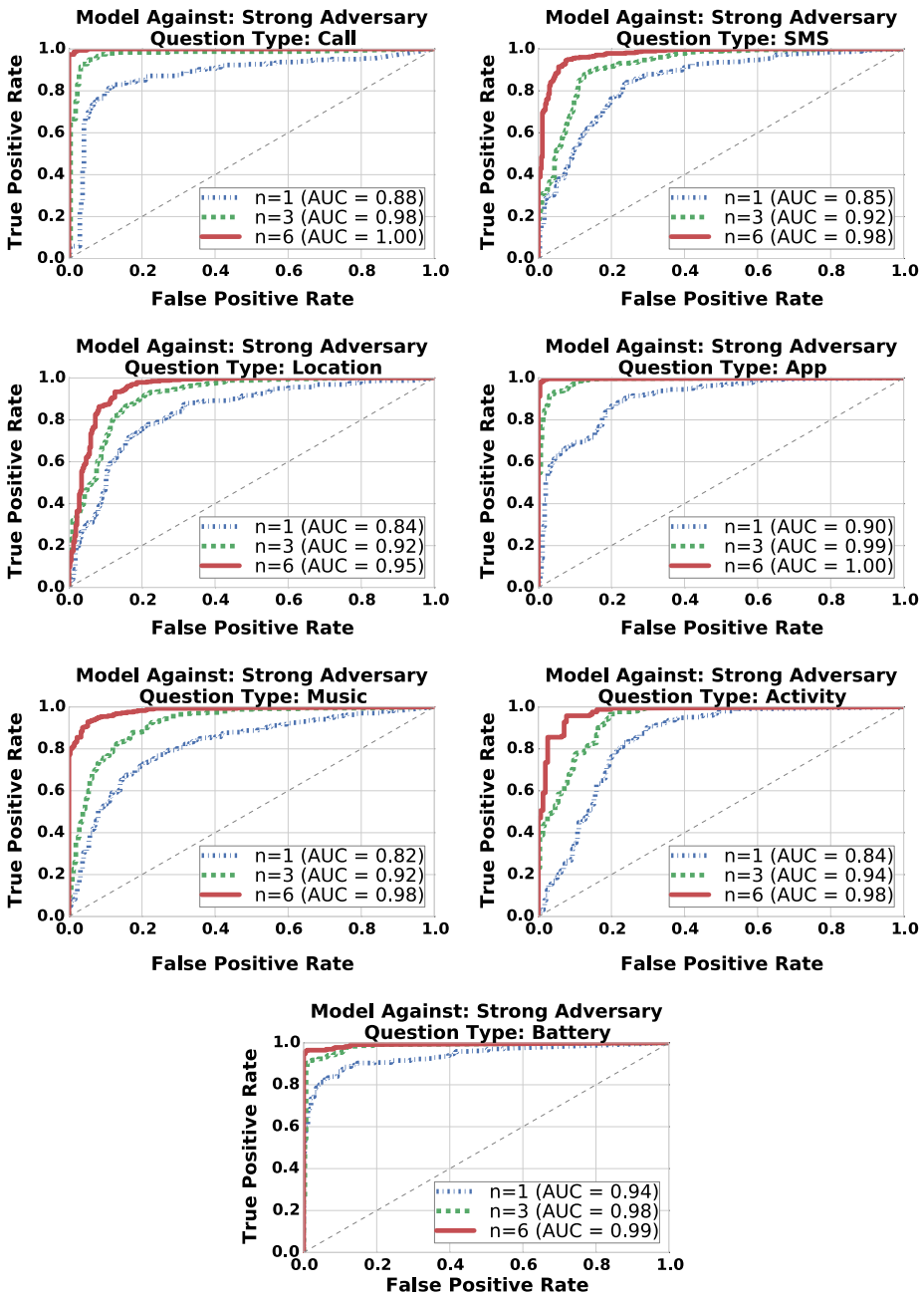

Fig. 7 Receiver operating characteristic (ROC) curves of the Bayesian classifier based scheme for different types of questions when modeled against naive adversaries for $n = 1, n = 3$, and $n = 6$ questions respectively

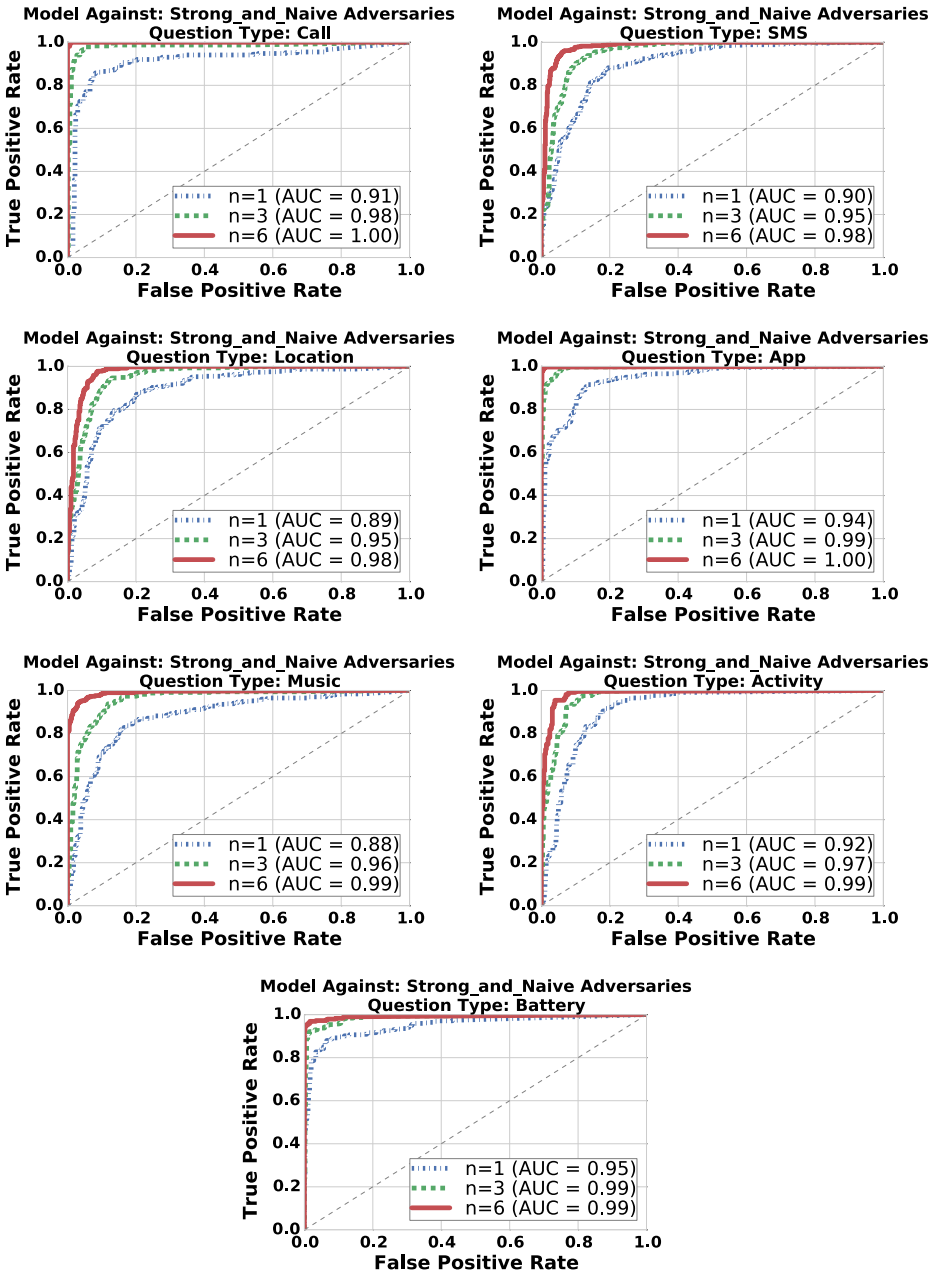

Fig. 8 Receiver operating characteristic (ROC) curves of the Bayesian classifier based scheme for different types of questions when modeled against strong adversaries for $n = 1, n = 3$, and $n = 6$ questions respectively

security questions. Specifically, in response to the question "would you consider using this system as a replacement of existing systems?", majority of the participants mentioned advantages of such systems over existing static challenge based systems. Some of the sample responses are below.

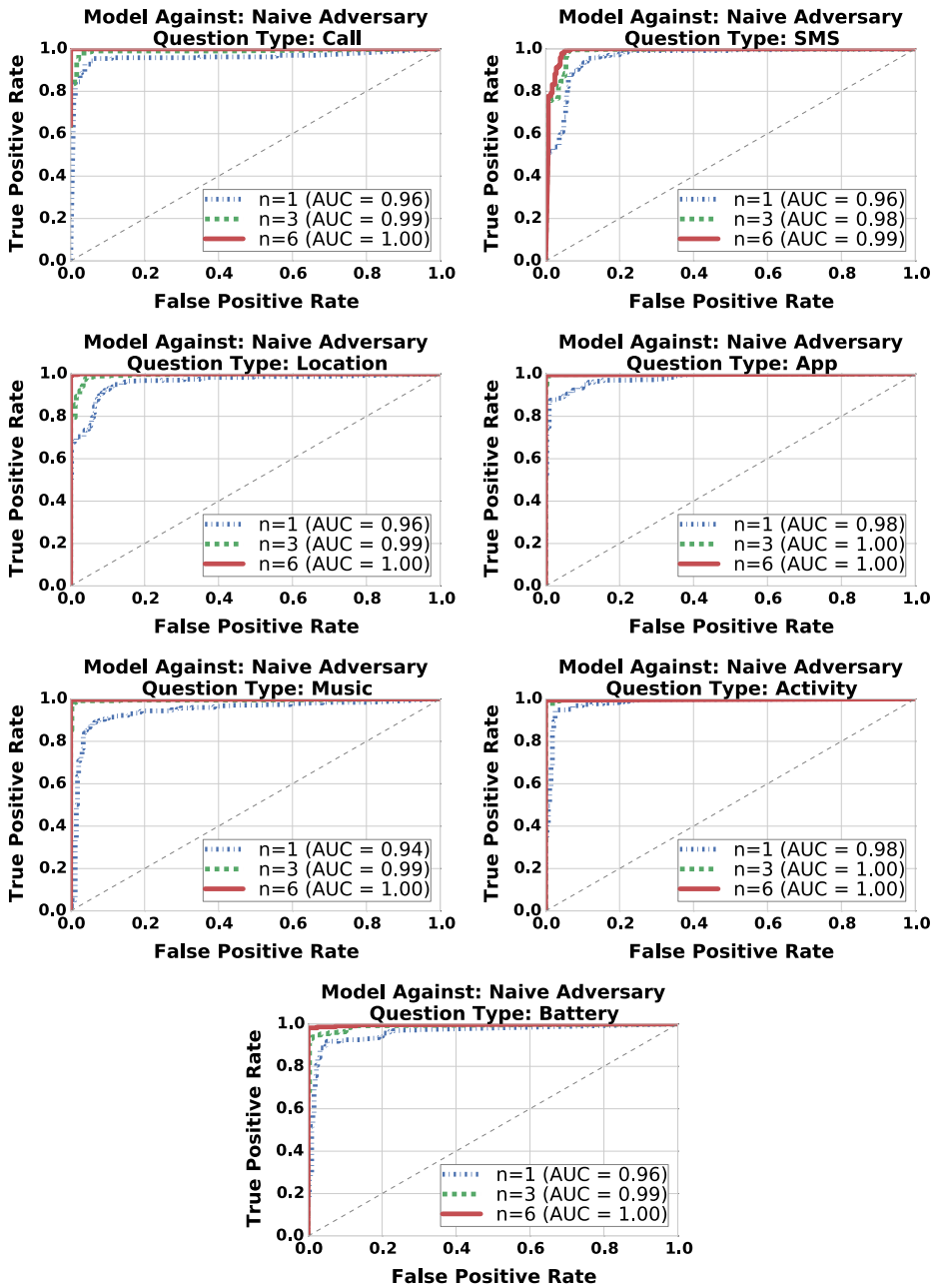

Fig. 9 Receiver operating characteristic (ROC) curves of the Bayesian classifier based scheme for different types of questions when modeled against strong and naive adversaries for $n = 1, n = 3$, and $n = 6$ questions respectively

Table 9 Classification performance of the Bayesian based classifier in identifying legitimate and adversarial users for 3 different attack scenarios (against naive adversaries, against strong adversaries, and against both strong and naive adversaries) for given question types for $n = 1$, $n = 3$, and $n = 6$ questions respectively

		Against naive				Against strong+naive				Against strong			
		TPR (%)	FPR (%)	F1 score (%)	Accuracy (%)	TPR (%)	FPR (%)	F1 score (%)	Accuracy (%)	TPR (%)	FPR (%)	F1 score (%)	Accuracy (%)
Call	n = 1	87.0	3.1	89.7	91.8	79.3	6.6	82.1	88.4	75.2	10.2	80.4	82.1
	n = 3	95.9	4.7	95.9	96.0	90.7	2.1	92.8	95.7	89.1	2.7	92.7	93.1
	n = 6	100.0	4.2	98.2	97.8	98.1	0.6	98.3	98.9	95.9	0.8	97.4	97.5
SMS	n = 1	76.8	5.0	82.3	86.2	69.7	10.7	73.7	81.6	65.1	14.3	72.1	74.5
	n = 3	93.4	4.4	94.7	94.1	88.1	8.2	87.4	90.2	86.2	12.6	87.4	86.8
	n = 6	97.8	4.0	97.3	96.8	94.2	5.5	92.3	94.3	92.8	7.8	93.1	92.6
Location	n = 1	84.3	7.1	86.0	89.1	75.1	14.4	73.4	82.3	70.8	17.0	74.0	77.0
	n = 3	96.3	3.2	96.8	96.3	93.2	10.5	88.3	90.7	88.7	16.7	87.3	85.8
	n = 6	99.2	0.0	99.6	99.5	97.1	7.7	92.1	93.8	95.2	14.5	91.8	90.4
App	n = 1	89.6	3.7	93.0	92.2	88.3	12.9	85.9	87.3	86.5	21.3	85.7	82.7
	n = 3	95.0	0.0	97.3	97.1	93.8	2.8	94.7	95.7	92.7	5.0	94.3	93.4
	n = 6	94.7	0.0	97.2	97.1	94.0	0.2	96.5	97.6	92.8	0.6	95.8	95.9
Music	n = 1	83.1	6.5	87.0	87.7	79.9	16.7	76.1	81.6	78.2	28.7	76.2	74.2
	n = 3	94.2	0.2	96.7	96.8	91.9	9.3	88.7	91.1	88.6	15.6	87.7	85.9
	n = 6	96.5	0.0	98.2	98.2	95.9	5.7	93.3	94.8	95.2	9.3	93.4	92.6
Activity	n = 1	93.3	6.2	93.4	93.2	87.2	17.2	81.5	83.9	83.1	27.1	81.5	78.1
	n = 3	95.9	0.0	97.8	98.0	93.5	7.4	90.8	92.6	91.4	17.1	90.1	86.8
	n = 6	96.5	0.0	98.2	98.0	95.5	4.0	94.4	95.6	93.7	7.6	94.2	93.2
Battery	n = 1	89.3	7.2	91.8	90.3	86.4	8.0	87.9	89.5	84.9	8.8	88.5	87.7
	n = 3	94.0	1.8	96.1	95.9	90.7	2.0	93.5	95.3	90.5	3.0	94.0	93.6
	n = 6	95.3	0.0	97.5	97.3	92.9	0.0	96.0	97.4	93.1	0.0	96.2	96.4

Four performance measures are shown: TPR, FPR, F1 score and accuracy

Table 10 User feedback on autobiographical authentication scheme

Statement	Call		SMS		Location		App		Music		Battery		Activity	
	Mode	Med	Mode	Med	Mode	Med	Mode	Med	Mode	Med	Mode	Med	Mode	Med
It was easy for me to recall	4	4	4	3.5	5	4	4	4	5	4	5	4	3	3
It was easy for my close friends to guess	1	2	1	2	3	3	3	3	1	3	2	2	2	2
It was easy for me to guess my close friends' questions	1	3	2	2.5	3	3	3	3	2	2	2	2.5	2	2
It was easy for a stranger to guess	1	1	1	1	1	1	1	1.5	1	1	2	2	1	1
It was easy for me to guess stranger's questions	1	1	1	1	1	1	1	2	1	1	2	2	1	1

A five-point Likert-scale was used on a scale of 1 (strong disagreement) to 5 (strong agreement) with the given statement. Mode and median values are shown for each statement and given question type

"- Yes, I think this is a very good system because I always have trouble resetting my password because I try to use different questions on different website so that if I get hacked, it's only that one website that gets hacked, and therefore I have trouble remembering the answers that I put for the questions. This proved to be very easy to guess my own questions but very difficult to answer questions about my pair and about a stranger, therefore I would feel very comfortable and protected using this system to protect my accounts."

"- Yes, I think that this system could potentially be much more secure than existing systems, as it would require either a very constant update on my behavior to successfully guess or a very thorough understanding of my daily habits, both of which I feel would be more difficult to easily identify than a single piece of information."

"- I would use this system to replace the old one. It is probably more secure than what is used now and does not require memorization. It seems easy to use and ..."

"-A personal anecdote: my ex-boyfriend knew all of my passwords while we were in a relationship and of course a lot of personal information (i.e., mother's maiden name, pet's name, etc), but when we broke up, he still had all of this information and could easily hack my accounts, so I had to go to ALL of my accounts and change the passwords. On the other hand, in the case of this system, he might be able to guess at my cell phone activities while we are together (not all, but most information). BUT, once the relationship is over, my cell phone usage would be significantly harder to guess for him as my activities and habits change much faster than personal questions such as pet name."

Despite the positive feedback, there were a few participants expressed their concerns regarding the usability aspects of the system. Some sample responses are below.

"-Sometimes it is hard for me to recall sms and call logs because I text so many different people."

"-Who you text/call, what apps you use and what music you listen to, it all changes a lot quite frequently in my life. So while probably way more secure, I would be worried that I would lock myself out of my accounts a lot."

Discussion and limitations of the study

In our study, we investigated the strengths, weaknesses, and usability aspects of different types of dynamic security questions. Our findings suggest that, while the raw accuracy varies for different question types, the **model-based accuracy** of the presented system exceeds the accuracy of static challenge question based systems significantly, and is also harder to compromise by adversarial users (e.g., low guessability) [4, 7, 8, 14].

However, while our study presents interesting findings, we would like to point out that all of the participants in our study were undergraduate students with age between 18 and 23 years. Thus, further studies are needed to investigate the difference between different types of user groups. Also, while the Bayesian classifier based model achieves

high accuracy, the model requires training data for both legitimate and adversarial users, which may not be always available in real-life. However, as we have shown in "Bayesian based classifier for authentication" section, one possible way to address this limitation is to train the model using a group of adversarial users' data that do not include any specific adversary, which is more likely to be available. For the requirement of training data from legitimate users, one may incrementally train the model by asking training questions and switch from training mode to prediction mode over time once the desired level of classification accuracy is achieved [46]. Other machine learning techniques such as SVM and feature extraction algorithms may be explored as well to improve the accuracy of the system.

Furthermore, based on our study, we identified several concerns that should be kept in mind while designing such systems. For instance, a hostile friend, who called/texted the target recently, may take advantage of this information to comprise the target's account if the authentication session consists solely of call and SMS based questions. Therefore, asking multiple and different types of dynamic security questions is essential to enhance the strength and security of such system. Also, as different kinds of information on the smartphone involves varying levels of sensitivity (e.g., contact names vs. list of app installed), poorly designed dynamic security question can reveal a user's privacy sensitive information over multiple iterations. For example, if phone call or SMS based questions are presented in multiple choice format, privacy sensitive information such as a user's contact names can be revealed/exposed to others through that question. Thus, when designing dynamic security questions, each data type should be evaluated thoroughly to ensure privacy. Finally, in our study, while most users who exhibit high smartphone usage behavior for all data types, some users may exhibit low smartphone usage behavior, making it harder to generate certain question types.

Despite these limitations, we would like to emphasize that the presented dynamic security question mechanism offers several advantages over static challenge question based system. For instance, in such system, users do not have to configure questions a priori. Also, unlike static security questions where many questions are often inapplicable for certain users (e.g., name of your first pet), questions in the presented system are customized for each user individually, making it harder to guess by mining online content.

Conclusions

In this paper, we presented the design and evaluation of dynamic security questions that are generated based on users' smartphone usage behavior and day-to-day activities captured by smartphones. Our findings suggest that the style of challenge questions and answer format can have a significant effect on user performance. Furthermore, while the performance of legitimate users turned out to vary across question types, a Bayesian based classifier can distinguish between legitimate and adversarial users with high accuracy. Finally, based on our exit survey, users were found to be positive in general towards the idea of using dynamic security questions for fallback authentication instead of static security questions.

Authors' contributions

YA designed, developed and implemented the smartphone-based fallback authentication system and conducted the experiments under the supervision of MK. YA analyzed data and YA and MK wrote the manuscript. MK designed the study and drafted the manuscript. All authors read and approved the final manuscript.

Acknowledgements
Part of this paper was published in our previous paper [25]. This work is supported by the National Science Foundation under Grant No. CNS-1251962. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agency.

Competing interests
The authors declare that they have no competing interests.

Received: 12 February 2016 Accepted: 3 July 2016
Published online: 05 September 2016

References

1. Florencio D, Herley C (2007) A large-scale study of web password habits. In: Proceedings of the 16th International Conference on World Wide Web. ACM, New York, pp 657–666
2. Shay R, Komanduri S, Kelley PG, Leon PG, Mazurek ML, Bauer L, Christin N, Cranor LF (2010) Encountering stronger password requirements: user attitudes and behaviors. In: Proceedings of the Sixth Symposium on Usable Privacy and Security. ACM, New York, p 2
3. Schechter S, Reeder, RW (2009) Measuring the comprehensibility of metaphors for configuring backup authentication. In: Proceedings of the 5th Symposium on usable privacy and security. ACM, New York, p 9
4. Bonneau J, Bursztein E, Caron I, Jackson R, Williamson M (2015) Secrets, lies, and account recovery: Lessons from the use of personal knowledge questions at google. In: Proceedings of the 24th International Conference on World Wide Web, WWW '15, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, pp 141–150. doi:10.1145/2736277.2741691. http://www.dx.doi.org/10.1145/2736277.2741691
5. Jakobsson M (2012) The death of the internet. Wiley-IEEE Computer Society Pr, New Jersey
6. Schechter S, Egelman S, Reeder RW (2009) It's not what you know, but who you know: a social approach to last-resort authentication. Proceedings of the SIGCHI conference on human factors in computing systems. CHI '09. ACM, New York, pp 1983–1992
7. Schechter S, Brush AB, Egelman S (2009) It's no secret measuring the security and reliability of authentication via secret questions. In: 30th IEEE symposium on security and privacy. IEEE, pp 375–390
8. Rabkin A (2008) Personal knowledge questions for fallback authentication: security questions in the era of facebook. In: Proceedings of the 4th symposium on usable privacy and security. ACM, New York, pp 13–23
9. Das S, Hayashi E, Hong JI (2013) Exploring capturable everyday memory for autobiographical authentication. In: Proceedings of the 2013 ACM International Joint Conference on pervasive and ubiquitous computing. ACM, New York, pp 211–220
10. Hang A, De Luca A, Hussmann H (2015) I know what you did last week! do you?: dynamic security questions for fallback authentication on smartphones. In: Proceedings of the 33rd annual ACM conference on human factors in computing systems, CHI '15. ACM, New York, pp 1383–1392
11. Hang A, De Luca A, von Zezschwitz E, Demmler M, Hussmann H (2015) Locked your phone? buy a new one? from tales of fallback authentication on smartphones to actual concepts. In: Proceedings of the 17th International Conference on human-computer interaction with mobile devices and services, MobileHCI '15. ACM, New York, pp 295–305
12. O'Gorman L, Bagga A, Bentley J (2004) Call center customer verification by query-directed passwords. Financial cryptography. Springer, Berlin, pp 54–67
13. Zviran M, Haga WJ (1990) User authentication by cognitive passwords: an empirical assessment. In: Information technology, 1990. 'Next Decade in Information Technology', Proceedings of the 5th Jerusalem Conference on (Cat. No. 90TH0326-9). IEEE, pp 137–144
14. Podd J, Bunnell J, Henderson R (1996) Cost-effective computer security: cognitive and associative passwords. In: Computer-human interaction. Sixth Australian conference on proceedings. IEEE, pp 304–305
15. Asgharpour F, Jakobsson M (2007) Adaptive challenge questions algorithm in password reset/recovery. First international workship on security for spontaneious interaction: IWIISI 7
16. Babic A, Xiong H, Yao D, Iftode L (2009) Building robust authentication systems with activity-based personal questions. In: Proceedings of the 2nd ACM workshop on assurable and usable security configuration. ACM, New York, pp 19–24
17. Dandapat SK, Pradhan S, Mitra B, Roy Choudhury R, Ganguly N (2015) Activpass: your daily activity is your password. Proceedings of the 33rd annual ACM conference on human factors in computing systems. CHI '15ACM, New York, pp 2325–2334
18. Nosseir A, Connor R, Dunlop M (2005) Internet authentication based on personal history-a feasibility test
19. Nosseir A, Terzis S (2010) A study in authentication via electronic personal history questions. In: Proceedings of the 12th international conference on enterprise information systems (ICEIS'10), vol. 5, pp 63–70
20. Nishigaki M, Koike M (2007) A user authentication based on personal history-a user authentication system using e-mail history. J Syst Cyber Inform 5(2):18–23
21. Albayram Y, Khan MMH, Bamis A, Kentros S, Nguyen N, Jiang R (2014) A location-based authentication system leveraging smartphones. In: Mobile data management (MDM), 15th international conference on IEEE, vol. 1, pp 83–88
22. Albayram Y, Khan MMH, Bamis A, Kentros S, Nguyen N, Jiang R (2015) Designing challenge questions for location-based authentication systems: a real-life study. Human-cent Comput Inform Sci 5(1):1–28
23. Gupta P, Wee TK, Ramasubbu N, Lo D, Gao D, Balan RK (2012) Human: creating memorable fingerprints of mobile users. In: Pervasive computing and communications workshops (PERCOM Workshops), International conference on IEEE, pp 479–482

24. Hang A, De Luca A, Hussmann H (2014) Using icon arrangement for fallback authentication on smartphones. CHI '14 Extended Abstracts on Human Factors in Computing Systems. CHI EA '14ACM, New York, pp 2467–2472
25. Albayram Y, Khan MMH (2015) Evaluating the effectiveness of using hints for autobiographical authentication: A field study. In: Eleventh symposium on usable privacy and security (SOUPS 2015). USENIX Association
26. Android's activity recognition API: recognizing the user's current activity. https://developers.google.com/android/reference/com/google/android/gms/location/ActivityRecognitionApi. Accessed 02 Feb 2016
27. Android fused location provider API. https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderApi. Accessed 02 Feb 2016
28. Conway MA (2009) Episodic memories. Neuropsychologia 47(11):2305–2313
29. The fake name generator. http://www.fakenamegenerator.com/order.php. Accessed 02 Feb 2016
30. Velho J, Marques D, Guerreiro T, Carriço L Physical intrusion detection and prevention for android smartphones
31. Top downloaded apps in Google Play Store. https://play.google.com/store/apps/top?hl=en
32. Spotify web API. https://developer.spotify.com/web-api
33. Ester M, Kriegel H-P, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. KDD 96:226–231
34. Sinnott R (1984) Virtues of the haversine. Sky Telesc 68:158–159
35. Google Maps Android API. https://developers.google.com/maps/documentation/android-api/. Accessed 02 Feb 2016
36. Thorpe J, MacRae B, Salehi-Abari A (2013) Usability and security evaluation of geopass: a geographic location-password scheme. In: Proceedings of the Ninth symposium on usable privacy and security, p 14
37. Kristo G, Janssen SM, Murre JM (2009) Retention of autobiographical memories: an internet-based diary study. Memory 17(8):816–829
38. Winkler WE (1990) String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage
39. Frary RB (1988) Formula scoring of multiple-choice tests (correction for guessing). Educ Meas Issues Pract 7(2):33–38
40. Rennie JD, Shih L, Teevan J, Karger DR et al (2003) Tackling the poor assumptions of naive bayes text classifiers. In: ICML, vol. 3. Washington DC, pp 616–623
41. Lewis DD (1998) Naive (bayes) at forty: the independence assumption in information retrieval. In: European conference on machine learning, Springer, Berlin, pp 4–15
42. Narayanan V, Arora I, Bhatia A (2013) Fast and accurate sentiment classification using an enhanced naive bayes model. In: International conference on intelligent data engineering and automated learning. Springer, Berlin, pp 194–201
43. Hastie T, Tibshirani R, Friedman J, Franklin J (2005) The elements of statistical learning: data mining, inference and prediction. Math Intell 27(2):83–85
44. Fawcett T (2006) An introduction to roc analysis. Pattern Recognit Lett 27(8):861–874
45. Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. Inform Process Manag 45(4):427–437
46. Kay M, Patel SN, Kientz JA (2015) How good is 85%? a survey tool to connect classifier evaluation to acceptability of accuracy. In: Proceedings of the 33rd annual ACM conference on human factors in computing systems. ACM, New York, pp 347–356