How Smart Your Smartphone Is in Lie Detection?

Md. Mizanur Rahman
Bangladesh University of Engineering and Technology
Dhaka, Bangladesh
engr.mizanbd@ymail.com

Atanu Shome
Bangladesh University of Engineering and Technology
Dhaka, Bangladesh
atanu.cse.ku@gmail.com

Sriram Chellappan
University of South Florida
Tampa, USA
sriramc@usf.edu

A. B. M. Alim Al Islam
Bangladesh University of Engineering and Technology
Dhaka, Bangladesh
alim_razi@cse.buet.ac.bd

ABSTRACT

Lying is a (practically) unavoidable component of our day to day interactions with other people, and it includes both oral and textual communications (e.g. text entered via smartphones). Detecting when a person is lying has important applications, especially with the ubiquity of messaging via smart-phones, coupled with rampant increases in (intentional) spread of mis-information today. In this paper, we design a technique to detect whether or not a person's textual inputs when typed via a smartphone indicate lying. To do so, first, we judiciously develop a smartphone based survey that guarantees any participant to provide a mix of true and false responses. While the participant is texting out responses to each question, the smartphone measures readings from its inbuilt inertial sensors, and then computes features like shaking, acceleration, tilt angle, typing speed etc. experienced by it. Subsequently, for each participant (47 in total), we glean the true and false responses using our own experiences with them, and also via informal discussions with each participant. By comparing the responses of each participant, along with the corresponding motion features computed by the smartphone, we implement several machine learning algorithms to detect when a participant is lying, and our accuracy is around 70% in the most stringent leave-one-out evaluation strategy. Later, utilizing findings of our analysis, we develop an architecture for real-time lie detection using smartphones. Yet another user evaluation of our lie detection system yields 84%-90% accuracy in detecting false responses.

CCS CONCEPTS

• **Human-centered computing** → *Ubiquitous and mobile computing design and evaluation methods*; *User studies*; *Ubiquitous and mobile computing design and evaluation methods*;

KEYWORDS

Lie detection, Human-Computer interaction, Android application, Machine learning, Ubiquitous Computing

ACM Reference Format:
Md. Mizanur Rahman, Atanu Shome, Sriram Chellappan, and A. B. M. Alim Al Islam. 2019. How Smart Your Smartphone Is in Lie Detection?. In *16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous), November 12–14, 2019, Houston, TX, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3360774.3360788

1 INTRODUCTION

Humans typically use textual, vocal and visual signs for communication. In doing so, they give out subtle cues from their facial expressions, tones, words used, body language and more, which experts use to gauge emotional state like depressive symptoms, anxiety, confidence, fear and also truthfulness (or lying). Even when humans attempt to mask reality by forcibly changing cues they give out, a trained expert can still see through these to gauge true emotional state.

In this context, the ability to detect when a person is lying is of paramount importance (and has been so for a very long time). In many aspects of daily life including police investigations, court trials, job interviews, and even in regular social communications, being able to detect lying has value. The most standard approaches today are based on a) polygraph tests that strap a person to a machine that measures changes in physiological signals like pulse rate, skin conductance, blood pressure etc. during lying; or b) a trained expert in body language to look for visual cues. Both these approaches are unsuitable for ubiquitous use, and also cannot be used when the communicating entities are not physically near, or not visible to each other during communication.

1.1 Background of This Study and Our Motivation

Since 1900, there has been an earnest interest to study deception, and particularly in the field of criminology. Benussi was the first researcher who worked on deception detection to the best of our knowledge [2]. He found changes in inspiration-expiration ratio, which was also confirmed by Burtt who also found changes in quantitative systolic blood-pressure during deception, also validated by Marston in [15] with a cohort of students and witnesses in court cases. Later, John Larson argued that Marston's method based on

MobiQuitous, November 12–14, 2019, Houston, TX, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-7283-1/19/11.
https://doi.org/10.1145/3360774.3360788

measuring intermittent blood pressure may miss out on detecting very brief episodes of deception, and accordingly, he initiated design of the modern lie detector or polygraph [8]. To do so, he modified the Erlanger sphygmograph to give a continuous measure of blood pressure and pulse rates. As we know, the polygraph is now widely used (sometimes as evidence during trails) for detecting lying, but it is still expensive, and requires significant expertise to operate.

Since it is believed that those that lie give our cues in their faces (e.g., appear more nervous [16]), Paul Ekman in the late seventies developed a Facial Action Coding System (FACS), and combined it with voice and speech measures to achieve 90% accuracy in detecting deception [4]. Depaulo and Morris later analyzed verbal and written outputs of liars to find distinctive patterns. They claim that liars usually take longer to start answering questions than truth tellers [16]. However, these techniques again require significant manual expertise and are not suitable for automation.

1.2 The Importance of Lie Detection in Cyber Space

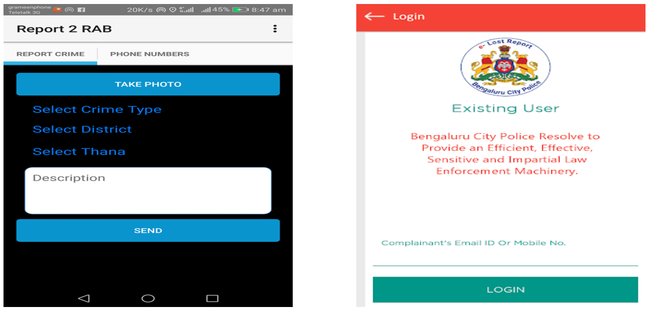
As of today, the rate of cyber communications is exploding. In the recent past, spread of misinformation on social media (some intentional and some unintentional) has become a major issue, since the impact of spreading lies to millions of users can have catastrophic consequences. Even during personal communication among friends on cyber space, identifying deception can be important. More recently, agencies in developing countries - for example, Rapid Action Battalion (RAB) in Bangladesh, and e-Lost Report and Police complaints in India, have created smartphone apps to let citizens contact law enforcement (Figure 1(a)-1(b)). Our brief interviews with related officers revealed they are seriously concerned about intentional false reports, and were very receptive to any technique that could determine if deception was used by any citizen when using these systems, that were designed for societal good.

Our motivation for the work in this paper, comes from the fact that today, smartphones are amongst the most preferred form of cyber communications. In parallel, modern smartphones come with a number of inertial sensors to detect motion. We want to determine if machine learning techniques can detect when a user is lying by detecting subtle changes in their typing patterns (when compared to being truthful) as measured by inertial sensors on smartphones.

1.3 Our Contributions

In this paper, our contributions are below:

- We develop a smartphone based survey incorporating a carefully chosen set of questions, which are guaranteed to invoke both true and false responses from participants. Each participant is expected to type out responses to each question, during which time the smartphone will unobtrusively record readings from its inbuilt inertial sensors. Using these readings, the smartphone computes features like degree of shaking, acceleration, tilt angle, typing speed etc.
- We then sort out responses of each participant to each question as true of false via informal interactions with them, and from our own prior social interactions with some of them. With this ground truth data, we implement several machine learning techniques to detect instances of lying based on features computed by the smartphone. Our results yield accuracies of 90%, 84% and 70% under same-user, cross-user and leave-one-out evaluation strategies respectively.
- Finally, we develop a dynamic end-user application, where anyone can create their own survey with self-selected questions, collect feedback from different participants, and check lying in the participant's answers. We let 42 users use this system. We find 88% accuracy with this system also, hence demonstrating the practicality of our system for widespread use.



Bengaluru City Police Resolve to Provide an Efficient, Effective, Sensitive and Impartial Law Enforcement Machinery.

(a) Report 2 RAB Android application

(b) e-Lost Report Android application

Figure 1: Existing smartphone based systems to report to law enforcement

2 RELATED WORK

Our work in this paper broadly falls in the category of human emotion detection using algorithmic techniques. We present important related work in this space.

In the early days, researchers started with detecting emotions from text-based communications [11, 14], using features like Keyword Spotting, Lexical Affinity Method, Learning-based Methods, and Hybrid Methods. Cheng et al., proposed a framework that identifies sentiment by computing opinion and lexica extracted from unlabeled textual data [9, 10]. In another related work, Liu et al., presented an emotion recognition method by extracting textual and non-textual features as applied to micro blogging [12, 13] data. To analyze the performance of Support Vector Machine (SVM) for sentiment analysis in Weka, two pre classified datasets of tweets are used and for comparative analysis, three measures, i.e., Precision, Recall and F-Measure are used.

Zeng et al. worked on emotion detection using multimodal fusion for human affect analysis including audiovisual fusion, linguistic and paralinguistic fusion, and multi-cue visual fusion based on facial expressions, head movements, and body gestures [5]. Ko et al. used hybrid deep learning algorithm for detecting emotion from facial expression [7].

Khanna et al., presented a method to recognize selected emotion categories from keyboard stroke patterns based on the significant difference between typing speed, frequency of using backspaces, and use of unrelated keys [6]. Gerald et al., proposed a method to detect stress-related changes in the behavior of individuals by using smartphones [1].

As we can see, recognizing emotions is an active area of HCI research, and recognizing emotions using data sensed by smartphones is an important sub discipline. To the best of our knowledge though, the issue of detecting lying from truthfulness using smartphone senors has not been attempted yet, and is our unique contribution in this paper.

3 DESIGN OF A USER SURVEY SYSTEM

We now present the design details of an Android application we developed that will prompt a participant to respond via textual inputs to a series of questions, some of which will elicit truthful responses, while others are guaranteed to elicit false responses.

3.1 Survey Application Development and Parameters Computed by Smartphone

Our Android application (Figure 2) contains a set of 45 questions, with one question appearing in each screen. For each participant in our study, we record his/her textual responses to each question. In parallel, the tri-axial accelerometer and gyroscope sensors in the smartphone will record readings as the participant is typing out responses. From the sensory data, the phone will compute the following parameters: average shaking, average acceleration, average tilt angle and average rotation computed in each millisecond. In addition, using available APIs, we enable the smartphone to compute the typing speed (i.e., number of keys pressed over time), the number of deleted characters, and the number of text suggestions used in each millisecond. Since computing all these parameters are straightforward, we do not emphasize their computations further in this paper. We wish to emphasize though that most smartphones today are equipped with these inertial sensors, and computing each feature is quick, unobtrusive and consumes very minimal energy.

Let L be the number of characters entered by a participant, T be total time taken to provide a complete answer, S be the total shake calculated by using Android library for a complete answer, A_c be the total acceleration calculated by using Android library for a complete answer, A_n be the total angle calculated by using Android library for a complete answer, R be the total rotation calculated by using Android library for a complete answer, D be the total number of deleted characters, S be the total number of suggestions used, T_s be Typing speed, A_s be the average shaking, A_a be the average acceleration, A_{an} be the average angle and A_r be the average rotation. Now, we calculate these as follows:

$$S = \frac{Total\ distance\ covered\ on\ X,\ Y\ and\ Z\ axis}{T} \tag{1}$$

$$A_c = \frac{Total\ distance\ covered\ on\ X,\ Y\ and\ Z\ axis - Gravity}{T} \tag{2}$$

$$T_s = \frac{L}{T} \tag{3}$$

$$A_s = \frac{S}{L} \tag{4}$$

$$A_a = \frac{Ac}{L} \tag{5}$$

$$A_{an} = \frac{An}{L} \tag{6}$$

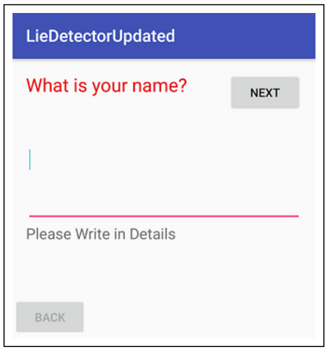


Figure 2: User interface of our application (used for both survey data collection and testing)

$$A_r = \frac{R}{L} \tag{7}$$

Note here that each question (among the 45 questions in total) comes in a separate screen (see Figure 2). After responding to one question, participants are then taken to the next screen. In this manner, the smartphone parameters computed can be synchronized with responses to each question (by comparing timestamps). Also, participant responses, smartphone parameters and timestamps are exported in the background to a server immediately after completing the survey. Naturally, this data is used for model development.

3.2 Selection of Questions for our Survey

One of the major challenges of this study is to identify a questionnaire for which participants will provide a reasonable mix of true and false responses. But, this is hard to do, and encompasses aspects related to participant age, gender, past experiences in life, cultural sensitivities and more. Upon extensive surveying of related literature on human psychology, we could not find insights on how to design questions for which participants are likely to lie, beyond a basic guideline that participants usually respond to sensitive questions with false answers [3]. While this was initially surprising (i.e., the lack of literature on how to make people lie), it was understandeable also, since the whole notion of lying, despite being common is also highly complex to predict or detect or rationalize.

As such, we decided to create our own questions based on our cultures, past social experiences, our own sensitivities, and basic common sense. At the outset, we decided to focus on a relatively homogeneous age group so that we minimize any age related diversities in deception. As such, the participants in our study were between ages 21 and 25 (more details are elaborated later). We point out that the questions created by us which we presumed will guarantee a false response from participants in this age group in our proposed study are in fact ones for which we authors would have most likely given false responses if we were the participants. In fact, after several informal discussions among the authors, and many more with our social contacts also in the age group identified above, we were confident that the questions we created will produce a judicious mix of true and false responses in our participant population.

Once the questions are identified, the issue next is to decide the order in which we want to present it to our participants to

Table 1: A partial set of survey questions (questions from the childhood to current age)

Survey questions
1. What is your name?
2. Where do you stay?
3. What is your date of birth?
4. What is your religion?
5. Where did you born and grow up?
6. What is the name of your primary school?
7. How and where was it?
8. Which type of work did you like in school?
9. When and how did you smoke for the first time?
10. Did you get caught by your parents while smoking? How?
11. Did you have any dream girl/boy? Who is s/he?
12. What did you spend your leisure time in college?
13. Which type of game did you like in college?
14. Did you tease any teacher in college?
15. Did you fall in love with any teacher?
16. When and how did you start a relationship?
17. What did you do with him or her?
18. What are your department and university?
19. Why did you choose the department?
20. What is your future plan?
21. What is the duration of your relationship?
22. Are you a virgin?
23. When and how did you make your first kiss?
24. Does your family know about your relation?
25. Will you marry your lover? Why?
26. When and why do you tease boys/ girls/ men/ women on a road or other areas?
27. What are your plans after your graduation?

get their responses. We had a few obvious choices: a) to present the non-sensitive questions first, and then the sensitive ones; b) to present the sensitive ones first, and then the non-sensitive ones; c) interleave the sensitive and non-sensitive questions; and d) decide order of questions randomly. Somehow, we felt that in the first two choices, participants will feel abrupt changes in emotion as they move from sensitive questions to the non-sensitive ones or vice versa, and we did not want that. In the case of interleaving them, or randomizing them also, we thought participants could lose engagement with our system beyond a point, since they do not see any pattern to the questions. What we instead decided was to order the questions that almost represents a linear narration about their recent past, while still interleaving sensitive and non-sensitive questions in between. In this manner, we felt that participants will see a pattern to our questions, that will bring back recent memories and they are more likely to be emotionally connected to our system when responding. In this manner, we design a survey with 45 questions in total, out of which 17 are sensitive and likely to yield false responses, while the other questions are non-sensitive and are likely to yield true responses. The partial set of questions (arranged in the form of a story capturing past history) is presented in Table 1. This is the order for all participants in our study.

4 CONDUCTING THE SURVEY AND ANALYSIS

4.1 Demographics and Initialization

We recruited 47 participants for our study in the age group of 21-25. Among these, 26 were male and 21 were female. Most of the participants are students, and some are in the IT industry. All of them are experienced users of smartphones. Most of the participants are familiar with at-least one author of this paper. We then presented a handout to each participant indicating to them that the purpose of this study was emotion detection. We did not specifically mention that detecting deception was the primary focus of the study.

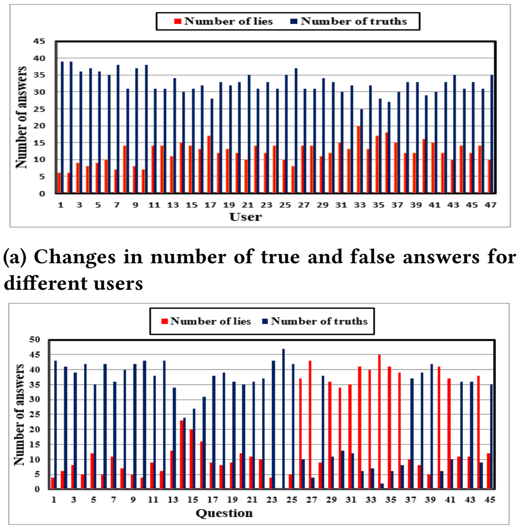
Participants enquired about confidentiality of their responses, and we assured them that their responses will be encrypted and stored in our server, and we assured them post completion of the study, all data will be deleted. Participants were also told that their identities are all anonymized. At this point, all participants willingly agreed to participate.

Participants were then given our smartphone with the app installed and they were escorted to a quiet room with a chair. Some of the participants sat, while some others stood and walked when responding to questions. Once the subject finished the survey, we met each person one to one and asked them casually if they did indeed give false responses to one or more questions, and it was indeed the case with all participants. Subsequently, we gave each participant another simple smartphone app (not shown here due to space limitations), wherein each participant was asked to choose the questions for which they answered truthfully, and the questions for which they did not answer truthfully. After (once again) getting confirmation from us on the confidentiality and anonymity of entered responses, each participant willingly entered this via the smartphone app, and they confirmed that they were completely truthful the second time. This information was immediately recorded in the server as well. All ids were anonymized, and we reiterate that the authors of this study did not access any textual response, nor did we know which user answered truthfully to which question.

4.2 Ground Truth and Feature Extraction

At the conclusion of our study, we collected a total 2115 responses from 47 participants. In Figure 3b, we plot trends to reveal the number of true and false responses against participants (anonymized) and questions in the survey. We present some interesting trends here. The number of false responses per participant ranged from a minimum of 3 to a maximum of 20, with a mean of 12 and a standard deviation of 3. In summary, the total number of true responses were 1539, and the total number of false responses were 576. As mentioned earlier, this data was provided by participants themselves. By correlating this data with appropriate time stamps of entered responses for each question, and the parameters computed in the smartphone, we determine the parameter values for true responses, and parameter values for false responses.

In Figure 4, we plot trends for five features - typing speed, average shaking, average acceleration, average tilt angle, average rotation against true and false responses for a subset of participants. As we see, there are interesting trends that are consistent across users. The average typing speed is lower (Figure 4a) for false



(a) Changes in number of true and false answers for different users

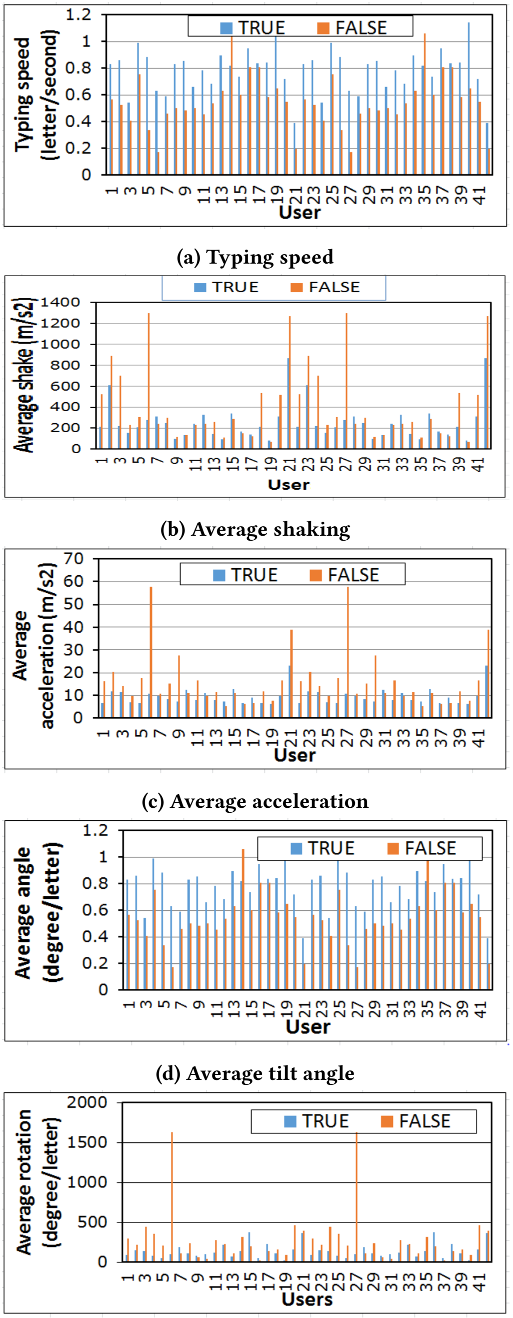
(b) Changes in number of true and false answers for different questions

Figure 3: Changes in numbers of true and false answers

responses compared to true responses. The average shake and average acceleration have the reverse trend as seen in Figures (4b & 4c). The average tilt angle is more for true responses, while the average rotation is more for false responses as seen in Figures (4d & 4e). We did not find any tangible differences in other features - number of suggestions used, number of deleted characters and touch pressure, and as such, we ignore them for model development. Only the five features plotted in Figure 4 will be used in our model.

4.3 Evaluation with Machine Learning Algorithms

Based on participant responses, we find that 576 answers are false and 1539 answers are true. To resolve this imbalance, we implement a Class Balancer algorithm. Class Balancer algorithm reweights the instances in the test data so that each nominal class, i.e., true and false, has the same total weight. The total sum of weights across all nominal instances will be maintained [17]. We then implemented several machine learning algorithms to evaluate how each algorithm leverages our five features identified above in detecting false responses. Table 2 presents results across several standard metrics for three evaluation strategies - same user; 10-fold cross validation; and leave-one-out. We see that same user evaluation results are consistently the best, since variations across users are ignored here. The leave-one-out strategy is the most stringent one, since testing data is completely unseen from training data, and the performance evaluation metrics are slightly poor here, since irrespective of which activity is performed, there are always subtle variations among different people when they do that activity, and these variations lower performance to a certain degree. But what we see is that in each evaluation strategy, the Random Forest algorithm outperforms other techniques. Random Forest techniques have the advantage of being extremely fast, efficient on big data and capable of overcoming overfitting. Figure 5 shows the parameters used in implementing Random Forest Algorithm. 3 shows the performance of machine learning algorithms on total data.



(a) Typing speed

(b) Average shaking

(c) Average acceleration

(d) Average tilt angle

(e) Average rotation

Figure 4: Variation in usages for different participants

5 EVALUATION ON AN INDEPENDENT COHORT, AND WITH A NEW SURVEY

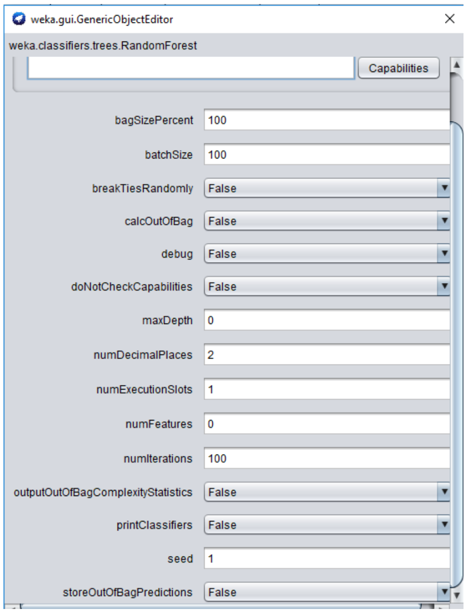
In order to evaluate our system via an independent group and with completely different questions, we recruited another cohort of 22

Table 2: Performance of machine learning algorithms in lie detection over our survey data

RC- Random Committee; RT- Random Tree; RF- Random Forest; LO-User validation leave one out; 10 Fold- 10 folds cross-validation;

	Accuracy in different algorithms									
	IB1		kStar		RC		RT		RF	
User	10 Fold	LO	10 Fold	LO	10 Fold	LO	10 Fold	LO	10 Fold	LO
1	85%	80%	85%	90%	83%	23%	90%	88%	85%	90%
2	93%	80%	95%	90%	90%	83%	95%	93%	93%	88%
3	75%	68%	80%	88%	75%	68%	80%	88%	80%	83%
4	73%	81%	83%	81%	88%	73%	83%	76%	83%	81%
5	74%	74%	69%	79%	80%	59%	85%	85%	77%	82%
6	73%	70%	80%	78%	88%	73%	88%	78%	88%	78%
7	74%	75%	75%	82%	78%	82%	85%	84%	79%	84%
8	96%	83%	96%	90%	98%	79%	98%	94%	96%	77%
9	66%	50%	73%	61%	77%	64%	86%	75%	77%	66%
10	74%	74%	81%	77%	77%	76%	86%	86%	84%	86%
11	69%	77%	73%	75%	65%	69%	77%	85%	81%	81%
12	57%	63%	54%	75%	54%	74%	57%	77%	49%	74%
13	79%	55%	73%	82%	79%	70%	73%	79%	70%	85%
14	58%	70%	67%	73%	67%	59%	73%	73%	79%	57%
15	53%	83%	57%	77%	53%	83%	73%	80%	70%	83%
16	58%	61%	71%	71%	58%	66%	66%	63%	61%	55%
17	84%	71%	81%	71%	84%	74%	81%	74%	84%	71%
18	81%	72%	78%	72%	66%	72%	88%	72%	81%	66%
19	75%	75%	72%	88%	72%	75%	78%	84%	78%	78%
20	80%	68%	80%	88%	76%	88%	80%	88%	80%	80%
21	72%	66%	72%	83%	76%	66%	79%	83%	69%	79%
22	89%	63%	89%	79%	95%	89%	95%	84%	89%	63%
23	75%	70%	75%	80%	70%	85%	70%	85%	75%	85%
24	64%	55%	58%	76%	67%	63%	70%	73%	58%	67%
25	42%	64%	49%	64%	49%	58%	55%	67%	33%	67%
26	79%	57%	75%	79%	75%	79%	82%	82%	75%	89%
27	61%	58%	58%	61%	61%	61%	66%	64%	61%	64%
28	76%	59%	66%	76%	72%	62%	76%	62%	72%	66%
29	70%	44%	74%	49%	72%	40%	77%	49%	67%	47%
30	74%	79%	74%	84%	68%	68%	84%	79%	74%	58%
31	65%	77%	65%	85%	65%	73%	69%	81%	62%	77%
32	81%	67%	90%	86%	86%	76%	86%	76%	86%	76%
33	63%	69%	63%	77%	71%	63%	66%	72%	69%	66%
34	62%	66%	72%	76%	59%	72%	59%	72%	66%	72%
35	75%	68%	78%	58%	75%	68%	78%	78%	73%	75%
36	67%	69%	61%	69%	64%	67%	78%	75%	64%	78%
37	72%	60%	70%	66%	68%	68%	74%	66%	66%	62%
38	89%	81%	84%	84%	84%	71%	92%	90%	89%	82%
39	80%	57%	73%	63%	77%	60%	80%	53%	77%	57%
40	70%	74%	75%	72%	70%	78%	79%	90%	76%	82%
41	72%	66%	69%	83%	72%	66%	72%	83%	75%	79%
42	65%	55%	70%	76%	62%	63%	65%	73%	54%	67%
43	72%	70%	80%	79%	87%	73%	88%	78%	86%	81%
44	74%	74%	80%	77%	77%	76%	86%	86%	85%	89%
45	80%	68%	80%	88%	76%	88%	80%	88%	82%	82%
46	69%	77%	75%	75%	65%	71%	77%	85%	81%	85%
47	72%	66%	72%	83%	76%	69%	79%	83%	81%	81%

new participants. None of these new participants were there in our original cohort of 47 participants. In the new cohort 13 were male and 9 were female. They were either students or IT professionals in the age group of 21-29. Each participant was once again briefed that


weka.gui.GenericObjectEditor

weka.classifiers.trees.RandomForest

bagSizePercent 100
batchSize 100
breakTiesRandomly False
calcOutOfBag False
debug False
doNotCheckCapabilities False
maxDepth 0
numDecimalPlaces 2
numExecutionSlots 1
numFeatures 0
numIterations 100
outputOutOfBagComplexityStatistics False
printClassifiers False
seed 1
storeOutOfBagPredictions False

Figure 5: Parameters of Random Forest algorithm

Table 3: Performance of machine learning algorithms in lie detection over our survey data

RC- Random Committee; RT- Random Tree; RF- Random Forest;

	Algorithms				
Measure	IB1	KStar	RC	RT	RF
True Positive Rate or Recall	76%	75%	76%	77%	78%
True Negative Rate	88%	94%	94%	89%	92%
Precision	91%	95%	95%	91%	94%
Negative Predictive Value	72%	68%	69%	72%	73%
False Positive Rate	12%	6%	6%	11%	7%
False Discovery Rate	9%	5%	5%	8%	6%
False Negative Rate	24%	25%	24%	23%	22%
Accuracy	81%	82%	82%	82%	83%
F1 Score	83%	83%	84%	83%	85%
Matthews Correlation Coefficient	64%	66%	67%	65%	68%
False Acceptance Rate	5%	3%	3%	5%	4%
False Rejection Rate	16%	18%	18%	17%	16%

the study was primarily for emotion detection. These participants also enquired about confidentiality and anonymity, and they were satisfied with our responses to protect their data/ identity. As before, the smartphone app was given to them and they entered into a room to answer all questions. This time, within the app, our Random Forest algorithm was implemented for real-time detection of false responses from true ones, and post completion of the survey, these results were encrypted and exported to our server.

Note that, the new survey was arranged chronologically also as in the earlier study. While some of the non-sensitive questions like asking for name and data of birth remained in the new survey, the sensitive questions created in the new survey were are different from the earlier one. The goal here is to see if our system is robust enough to operate on unseen participants, and for unseen questions/ responses. The number of questions in this survey were 35. A sample of questions are presented in Table 4.

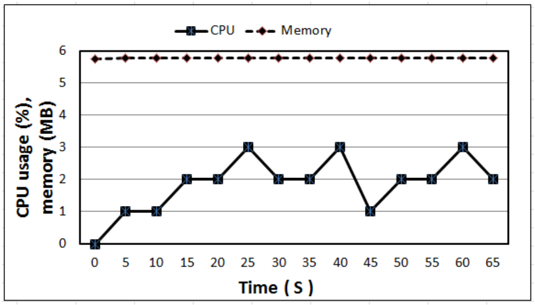


Figure 6: Resource usages by our application

Table 5 presents our results. We obtain these results by cross checking the output of our Random Forest algorithm within the app with each participant for validation. In this new study, we collected a total of 770 responses. Among these 498 answers were true and 272 answers were false as identified by our system in real-time. After validation from the 22 participants, we found that there were a total of 483 true responses and 287 false responses which is the ground truth. Table 5 presents our results across several metrics with very favorable results. The overall accuracy is 96% in detecting false responses which we believe is impressive.

We also present CPU and memory usage as shown in Figure 6, to demonstrate that resource utilization by our real-time prediction application is small.

6 PROPOSED ARCHITECTURE

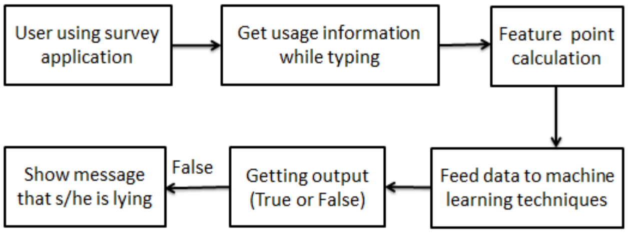
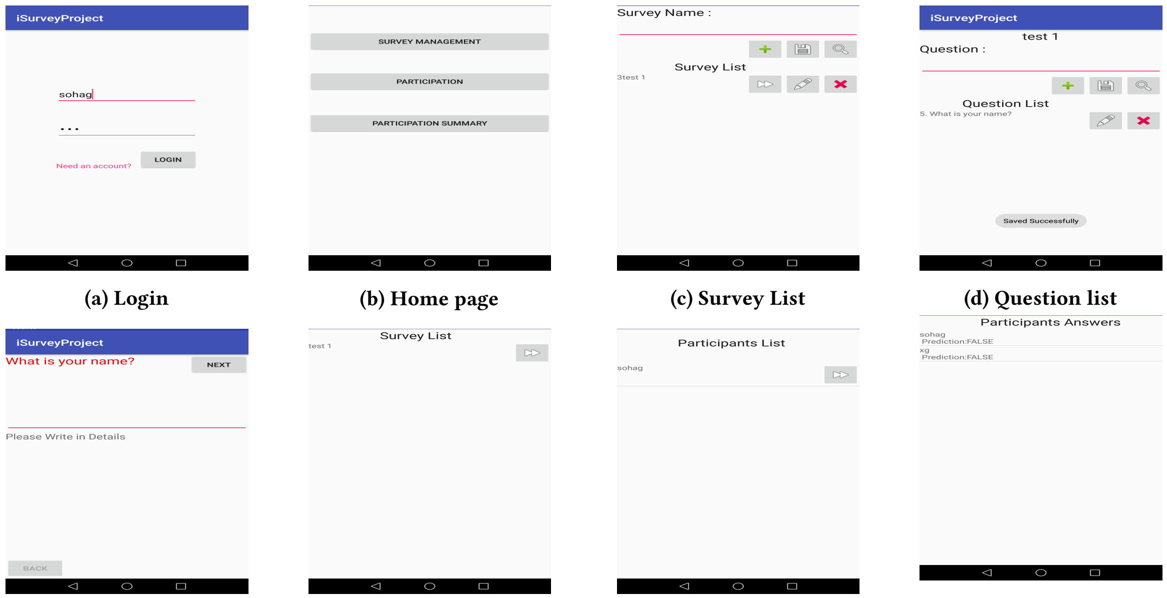

User using survey application
Get usage information while typing
Feature point calculation
Feed data to machine learning techniques
Getting output (True or False)
False
Show message that s/he is lying

Figure 7: Proposed architecture to detect a lie

As the results of our conducted survey using our developed survey system provides a good performance on identifying True and False answers, we propose an architecture for lie detection using smartphones. Figure 7 presents our architecture for lie detection using smartphones. Here, when participants provide their answers in our application, we collect several usage information in parallel. Afterwards, we extract features from the usage information (as pesented in Section 4) and feed them to the machine learning algorithm for lie detection. If the result is detected as false then the system returns a message that the participant is lying.



(a) Login (b) Home page (c) Survey List (d) Question list

(e) Answer submission page (f) Survey list in summary (g) Participants list in summary (h) Prediction summary for individual participant

Figure 8: Different steps and snapshots of our end-user application for real-time testing

7 APPLICATION DEVELOPMENT BASED ON OUR PROPOSED ARCHITECTURE

Using the proposed architecture, we develop a dynamic application. The objective of developing this application is to find out actual accuracy in real-life surveys, where a user at own can set the questions to be asked to a new participant. To do so, we develop our application in such a manner so that anyone can create a customized survey, collect the data, and check the summary. In the summary, our solution will present which responses are true and which are not.

7.1 Dynamic Application Development and Its Performance Evaluation

We develop an Android application where a user can create own account and then access the system after login (Figure 8a). Three options are shown in the home page therein (Figure 8b), where the user can manage survey, conduct the survey, and check summaries of a conducted survey. In our survey management, the user can create surveys (Figure 8c) and then add questions (Figure 8d), which can be subsequently presented to the participants for their responses. After setting all the questions under a survey, a participant can start participating in the survey through providing own feedback (Figure 8e). When the user completes getting feedback from the participants, the user can see a summary of answers with corresponding detection statuses determined by our solution for each participant participated in the survey (Figure 8f - 8h). Using our developed application, a user is free to conduct multiple surveys with own-selected questions. We present this application to a set of users for field testing.

Note that the own-selection of questions in this part of study is introduced to address the potential issue of biasness in our earlier parts. For example, it may be queried in our earlier parts of this study that whether our extracted outcomes came from truthfulness or other aspects such as toughness of questions. As this part of study gets conducted with diversified questions set by the users (not by us), such biasness is expected to be neutralized.

7.2 Demography of the Survey Participants

We provided our application to new 30 users (no overlapping among users of this survey and the earlier surveys) recruited voluntarily. They set the own questions and collected feedback over 30 days from 42 participants, where 24 were male and 18 were female. Most of the participants are students, some are software engineers, and the rest are from different occupations. Accordingly, most of the participants are from the age range of 21-25 years having prior experience on smartphone usage. Besides, most of the participants answer the survey questions in a state (mostly sitting) as similar to the earlier.

7.3 Performance Evaluation Results

We collect feedback from the 30 volunteers on our application. Here, 42 participants provided their 420 answers (295 were true and 145 were false) at their own. We find an average of 84% accuracy (Table 6 presents further detail) over this data having real time prediction with diversified questions. This result again demonstrate a great potential of lie detection using smartphones.

8 APPLICABILITY OF OUR RESEARCH

Smartphone-based complaint system such as Report 2 RAB, Police Helpline BD, Bangladesh Bank Complaint, e-Lost Report and police complaints in India, Police Complaint App in Ocland etc. can use

Table 4: Second questionnaire used in our study for real-time lie detection

Survey questions
1. What is your name?
2. What is your date of birth?
3. What is your educational status?
4. What is your religion?
5. What is your job/occupation?
6. What is your greatest weakness?
7. What are your strengths?
8. What are you most proud of?
9. What is your greatest fear?
10.What do you like to do?
11. Tell me about your worst boss.
12. How would you deal with a high-strung personality?
13. Where do you see yourself in five years?
14. Tell me about a project or work you worked on that required heavy analytical thinking
15. Can you describe a time when your work was criticized?
16. What was the most difficult period in your life, and how did you deal with it?
17. Tell me about a time you faced an ethical dilemma.
18. How do you want to improve yourself in the next year?
19. What are your lifelong dreams?
20. What do you ultimately want to become?
21. Do you pray regularly? What about your family members?
22. What is the punishment in your religion for avoiding your prayer?
23. What is the opinion about corruption in your country?
24. What do you do if someone offers you hush money (money that is paid so that someone will not tell other people about embarrassing or illegal behavior or work)?
25. What is the punishment for taking hush money in your religion?
26. What is your opinion about usury/interest system in your country?
27. Did you ever get usury/interest money? What do you do with banking interest?
28. What is the punishment for taking usury/interest money in your religion?

Table 5: Performance of our application developed for real-time lie detection using Random Forest

Measure	Random Forest
True Positive Rate or Recall	100%
True Negative Rate	90%
Precision	94%
Negative Predictive Value	100%
False Positive Rate	10%
False Discovery Rate	6%
False Negative Rate	0%
s Accuracy	96%
F1 Score	97%
Matthews Correlation Coefficient	92%
False Acceptance Rate	4%
False Rejection Rate	0%

Table 6: Performance evaluation of real-life testing of our solution using Random Forest

Measure	Random Forest
True Positive Rate or Recall	91%
True Negative Rate	74%
Precision	86%
Negative Predictive Value	83%
False Positive Rate	26%
False Discovery Rate	14%
False Negative Rate	9%
Accuracy	84%
F1 Score	88%
Matthews Correlation Coefficient	67%
False Acceptance Rate	10%
False Rejection Rate	6%

our approach for detecting lies. Through our application, the authority can determine true issues with high accuracy from a large number of submitted complaints. Again, social media such as Facebook, Twitter, LinkedIn, Google+ etc. can also use our approach for reducing the number of false content updates uploaded by their users. E-recruitment systems can also be a part of the applicability of our research. As nowadays a large number of fake applications are submitted in our online recruitment applications. Our approach can reduce this tendency from the applicants through identifying lying. Nonetheless, usage at the micro-level, for example in police interrogation or even questioning by parents to children at the family level can be other avenues of application of our study. Note that privacy might be an issue in some of applications (for example social media applications). We left study on this issue as our future work.

9 DISCUSSION

In this research, we investigate the task of lie detection using a smartphone. While collecting data and testing the performance of our application in real-life, we use smartphones of different brands such as Samsung, Huawei, xiaomi, Walton, Symphony etc. to check the impact of using different smartphones. We find a good level of accuracy in all smartphones. Besides, we initially presented sensitive and non-sensitive question either at random sequences or in sequence having two disjoint groups. For such presentation, participants provided only boolean answers or skip the questions. However, in the case of our story-like presentation of the questionnaire, the participants provided valid answers. This indicates that participants avoid completely random or isolated questions, since they are not engaged under such cases.

10 CONCLUSION AND FUTURE WORK

Existing lie detection systems generally demand high cost, infrastructural overheads, or hard-to-find domain experts. These systems also frequently require an interviewer to ask questions to participants. Thus, existing systems for lie detection are far from being a ubiquitous solution. Even though smartphones have been widespread in use in many parts of the world today, it is yet to be investigated whether the smartphones can be used for the purpose of lie detection. As smartphones are generally equipped with many sensors, there remains a high chance of using them in lie detection. However, to the best of our knowledge, such a smartphone sensor-based lie detection mechanism has not been focused in the literature till now.

In this study, we propose a new mechanism for detecting lying using smartphones. To do so, we design a customized survey system having a judiciously chosen set of questions that provokes participants to provide both true and false responses. We collect responses and corresponding usage data from 47 users through the survey system. Subsequently, we analyze the collected data using several machine learning algorithms and find that Random Forest can classify true and false responses over our collected data with a high accuracy of 83%.

Afterwards, based on our findings, we develop a new application that can detect the nature of a response, i.e., whether true or false, just after providing the response. We conducted survey using this new application over 22 participants using a new set of questions. We find that the application can provide an average of 96% accuracy.

Subsequently, we develop a dynamic application where a user can set his/her own questions and present that to participants requesting their responses. We present the application to 30 users who conducted survey over new 42 participants using their own set of questionnaires. We find 84% accuracy in this dynamic real-life end-user testing. All these experiments exhibit a robust prospect of smartphones to be used to detect falsification in the digital space, when the lying episode actually happens, and naturally can have many innovative applications.

It is worth mentioning that the participants in our study were common people. It is yet to be investigated whether an expert liar could be detected during lying using our solution. Such an investigation needs more rigorous surveys involving expert liars, which remains a future work of this study. Nonetheless, lie detection under different operational states (for example under stress) and emotional states (for example under anger) remain yet another avenue of our future research.

ACKNOWLEDGEMENT

This work was supported in part by US National Science Foundation (Grant # 1718071). Any opinions, findings and conclusions are those of the authors alone, and do not reflect views of the funding agency.

REFERENCES

[1] Gerald Bauer and Paul Lukowicz. 2012. Can Smartphones Detect Stress-Related Changes in the Behaviour of Individuals? *Work in Progress session at PerCom* (2012), 423 – 426.

[2] Benussi. 1914. On the Effects of Lying on Changes in Respiration. *Archiv fur die Gesamte Psychologie* (1914).

[3] Daniel Corstange. 2008. Sensitive Questions, Truthful Answers? Modeling the List Experiment with LISTIT. *Political Analysis* 17, 1 (December 2008), 45 – 63.

[4] P. Ekman and W. Friesen. 1978. Facial Action Coding System: A Technique for the Measurement of Facial Movement. *Consulting Psychologists Press* (1978).

[5] Zhihong Zeng ; Maja Pantic ; Glenn I. Roisman ; Thomas S. Huang. 2009. A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. IEEE, 39 – 58.

[6] Preeti Khanna and M.Sasikuma. 2010. Recognising Emotions from Keyboard Stroke Pattern. *International Journal of Computer Applications* 11, 9 (December 2010), 1 – 5.

[7] Byoung Chul Ko. 2018. A Brief Review of Facial Emotion Recognition Based on Visual Information. In *US National Library of Medicine National Institutes of Health*. IEEE, 401.

[8] John A. Larson. 1921. Modification of the Marston Deception Test. *Journal of Criminal Law and Criminology* (1921).

[9] Zheng Lin, Xiaolong Jin, and Xueqi Cheng. 2014. Make It Possible: Multilingual Sentiment Analysis without Much Prior Knowledge. *IEEE/ WIC/ ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)* (2014), 79 – 86.

[10] Zheng Lin, Songbo Tan, and XueQi. 2011. Language-independent Sentiment Classification Using Three Common Words. *CIKM* (October 2011), 24 – 28,.

[11] Chun-Chieh Liu, Ting-Hao Yang, Chang-Tai Hsieh, and Von-Wun Soo. 2009. Towards Text-based Emotion Detection: A Survey and Possible Improvements. *International Conference on Information Management and Engineering* (2009).

[12] Shenghua Liu, Fuxin Li, and Xueqi Cheng. 2013. Adaptive Co-Training SVM for Sentiment Classification on Tweets. *CIKMâĂŹ13* (Oct 2013), 2079 – 2088.

[13] Shenghua Liu, Wenjun Zhu, and Xue qi Cheng. 2013. Co-training and Visualizing Sentiment Evolvement for Tweet Events. *WWW 2013 Companion* (May 2013), 13 – 17.

[14] C. Maaoui, A. Pruski, and F. Abdat. 2008. Emotion recognition for human machine communication. *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 08)* (Sep 2008), 1210 – 1215. https://doi.org/10.1109/IROS.2008.4650870

[15] E. H. Marstonr. 1920. Physiological Possibilities in the Deception Test. *Journal of American Institute of Criminal Law and Criminology* (1920).

[16] B. M. DePaulo W. L. Morris. 2004. Discerning lies from truths: Behavioural cues to deception and the indirect pathway of intuition. *The detection of deception in forensic contexts* (2004).

[17] Weka. [n. d.]. Class Balancer. Retrieved Febuary 14, 2018 from http://weka.sourceforge.net/doc.dev/weka/filters/supervised/instance/ClassBalancer.html.