M.Sc. Engg. Thesis

DEVISING A UBIQUITOUS SOLUTION FOR LIE DETECTION

by

Md. Mizanur Rahman (1014052051 F)

Submitted to

Department of Computer Science & Engineering

(In partial fulfillment of the requirements for the degree of Master of Science in Computer Science & Engineering)

Department of Computer Science & Engineering

Bangladesh University of Engineering & Technology (BUET)

Dhaka 1000

March 10, 2018

Dedicated to my loving parents

Author's Contact

Md. Mizanur Rahman

Email: `engr.mizanbd@ymail.com`

The thesis titled "DEVISING A UBIQUITOUS SOLUTION FOR LIE DETECTION", submitted by Md. Mizanur Rahman, Roll No. **1014052051 F**, Session October 2014, to the Department of Computer Science & Engineering, Bangladesh University of Engineering & Technology, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Master of Science in Computer Science & Engineering and approved as to its style and contents. Examination held on March 10, 2018.

Board of Examiners

1. ______________________

Dr. A. B. M. Alim Al Islam — Chairman (Supervisor)
Associate Professor
Department of Computer Science & Engineering
Bangladesh University of Engineering & Technology, Dhaka.

2. ______________________

Prof. Dr. Md. Mostofa Akbar — Member (Ex-Officio)
Head and Professor
Department of Computer Science & Engineering
Bangladesh University of Engineering & Technology, Dhaka.

3. M. M. Islam

Prof. Dr. Md. Monirul Islam — Member
Professor
Department of Computer Science & Engineering
Bangladesh University of Engineering & Technology, Dhaka.

4. Rifat

Dr. Rifat Shahriyar — Member
Assistant Professor
Department of Computer Science & Engineering
Bangladesh University of Engineering & Technology, Dhaka.

5. Nova Ahmed

Dr. Nova Ahmed — Member (External)
Associate Professor
Department of Electrical and Computer Engineering
North South University, Dhaka.

Candidate's Declaration

This is hereby declared that the work titled "DEVISING A UBIQUITOUS SOLUTION FOR LIE DETECTION", is the outcome of research carried out by me under the supervision of Dr. A. B. M. Alim Al Islam, in the Department of Computer Science & Engineering, Bangladesh University of Engineering & Technology, Dhaka 1000. It is also declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Md. Mizanur Rahman
Candidate

Acknowledgment

First of all, I would like to express my heart-felt gratitude to my supervisor, Dr. A. B. M. Alim Al Islam, for his constant supervision of this work. He helped me a lot in shaping, deciding steps of my work, and providing infrastructural supports.

I would also want to thank the honorable members of my thesis committee: Prof. Dr. M. Sohel Rahman, Prof. Dr. Md. Monirul Islam, Dr. Rifat Shahriyar, and specially the external member Dr. Nova Ahmed, for their encouragements, insightful comments, and valuable suggestions.

I am also thankful to volunteers (students and the officials from different universities and other organizations) who helped me to continue my research. I am also grateful to all honorable teachers of the department for their comments and suggestions.

Last but not the least, I remain ever grateful to my beloved parents and family members for their inspirations behind every success of mine.

Abstract

Lying, as always, remains a significant part of our day to day interactions covering both physical communication and digital communication using devices such as smartphones. However, to the best of our knowledge, an effort is yet to be made to detect lying utilizing the ever increasing capabilities of smartphones. Therefore, in this paper, we investigate how far we can go in detecting lying through exploiting smartphones. To do so, first, we judiciously develop a set of questionnaire that guarantees to indulge a person in providing a mix of true and false responses. Here, we develop a survey system worth of deploying in smartphones. The system, along with collecting the responses, accumulates corresponding usage data such as shaking, acceleration, tilt angle, etc. while holding the smartphone. Subsequently, after distinguishing false responses from true ones based on informal communication and other verifications, we present distinguished responses and corresponding usage data collected from 47 participants to several machine learning algorithms. We find that we can achieve from 72% to 81% accuracy in identifying false responses through analyzing the usage data using machine learning algorithms. Later, utilizing findings of this analysis, we develop two different architectures for real-time lie detection using smartphones. Yet another user evaluation of the developed and implemented architectures confirms 84%-90% accuracy in lie detection.

Acronyms List

FAR = False Acceptance Rate

FRR = False Rejection Rate

RF = Random Forest

RT = Random Tree

RC = Random Committee

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Individuals generally impart utilizing verbal, vocal, and visual signs for communicating each other. The words they pick, their voice quality, and various body signals - all give data about their passionate and intellectual state that might indicate whether they are lying. The lie catcher needs to see and decipher these complex and extremely unpretentious prompts. Expert lie catchers substantially contrast from the (frequently misinformed) novice by the signs they search for, the trust they have in them, and the way they decipher. On the other, liars generally attempt to spill double-dealing. Most of them make decent attempts to conceal their duplicity, however, it is troublesome endeavoring to control own words, voice, face, feet, and hands all in the meantime. Here, their voices and faces often remain undeniable in conveying critical signals. Thus, these can often aid in lie detection.

Lie detection is one of the major areas of applied psychology. Lie detection becomes essential to identify whether someone is lying in police interrogation, court trials, border control interviews, intelligence interviews, social communication, interviews, and so on. To identify lying, many of the psychologists and practitioners have developed lie detection equipment and techniques. This type of equipment evaluates behavioral observation, analyze speech, and measure peripheral physiological responses to record brain activity. On the other hand, an expert is required to detect lying using psychological techniques. However, in both cases, there remain generally substantial amounts of true negative and false positive cases in such lie detection.

1.1 Background of This Study and Our Motivation

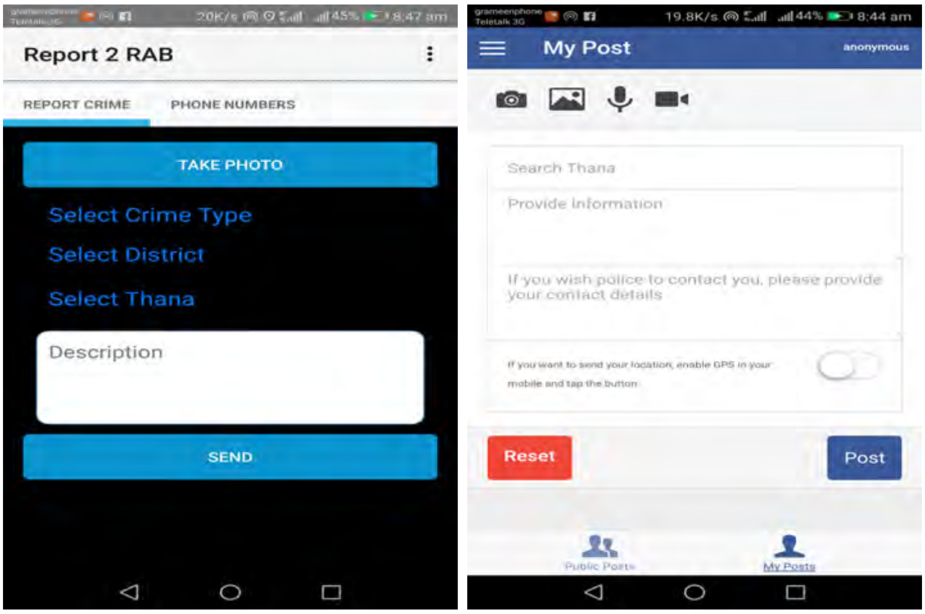
The study on lie detection using psychological changes in human body started in early 1900s. Benussi was the first researcher who worked on deception detection to the best of our knowledge [1]. He found changes in inspiration-expiration ratio, which was also confirmed by Burtt. Burtt conducted subsequent studies and found changes in quantitative systolic blood-pressure. Marston conducted his research on blood-pressure and detected an increase in systolic blood pressure with 90-100% accuracy [2]. The participants of his research were students and witnesses of court cases. Later, John Augustus Larson criticized Marstons method based on intermittent blood pressure, as with brief changes in emotion, it could be lost. Accordingly, he invented modern lie detector or polygraph [3]. To adjust this technique, he modified the Erlanger sphygmograph to give a continuous blood pressure and pulse curve and used it to study 4,000 criminals.

Besides, Depaulo and Morris found that those who are lying appear more nervous than those telling the truth [4]. Paul Ekman used Facial Action Coding System (FACS) and combined voice and speech measures to reach up to a detection accuracy of 90% [5]. However, there is currently no evidence to support such claim, as similar measures may appear in truthful emotional states. Depaulo and Morris studied on verbal and written output of liars to find distinctive patterns and they found that liars mostly take longer to start answering questions than truth-tellers [4]. However, this can also happen in truthful cases. Nonetheless, one thing in common to all these studies is that the findings are dependent on subjects and demands highly expert deciphering personnel. Thus, these findings are yet to be adopted to detect lie in cyber spaces. Moreover, to the best of our knowledge, no research study on lie detection in cyberspace has been conducted till now.

In recent days, communication through cyberspace has got very popular due to its easy access. Cyberspace is mainly accessed by using smartphones, notebooks, laptops, etc. Usage of smartphones has experienced a rapid growth in recent times due to its portability, cost, and features [6]. Users can get connected to the Internet through mobile networks and WiFi services exploiting smartphones. Thus, interaction with family, friends, and others through social media access using smartphones has increased sharply. Smartphones, providing such accesses, are generally equipped with different sensors such as motion sensor, touch sensor, etc.

These sensors can provide us different data and we can get several usage behaviors by using these sensors. If a mechanism of lie detection using the sensors available in a smartphone could be devised, the solution can be used for lie detection in many cases. However, to the best of our knowledge, devising such a mechanism for lie detection using smartphones is yet to be focused in the literature.

Recently, many government agencies (for example, Rapid Action Battalion (RAB) in Bangladesh) started a new complaint system over smartphones using an Android application (Figure 1.1a and 1.1b). Using such an application, a citizen can complain against any person providing the images and textual data. However, it is now near to impossible to determine if the complain is true or intentional, as there is now no solution available for such determination.



If you wish police to contact you, please provide your contact details

If you want to send your location, enable GPS in your mobile and tap the button

(a) Report 2 RAB Android application (b) BD Police Helpline Android application

Figure 1.1: Crime or complain reporting systems available in Bangladesh

1.2 Objectives of This Thesis

In this thesis, we will develop a solution for lie detection using smartphones. Here, information will be collected using sensors of a smartphone while providing true and lie responses through the smartphone. Afterward, based on outcomes of an analysis over the collected information, we will propose a new lie detection approach. Thus, the main objectives of this thesis comprise the following:

i. To propose a ubiquitous solution for detecting lie using smartphones.

ii. To develop a survey system in a smartphone for collecting both true and false responses along with storing corresponding sensors' data.

iii. To analyze the collected data and present them to different machine learning algorithms to identify the best machine learning algorithm for lie detection.

iv. To develop and implement standalone real-time solutions for lie detection in smartphones through integrating the best-found machine learning algorithm.

v. To perform experimentation with the developed standalone solution.

1.3 Our Contributions

Based on this study, our contributions are as follows:

- We develop a survey system incorporating a carefully chosen set of questionnaire, which can invoke both true and false responses from participants. The system logs various usage parameters in addition to storing the responses.

- We collect usage data and feedback from 42 users and enable several machine learning techniques to detect lying over the collected data after labeling true or false responses through informal communication and other verification. We achieve up to a good level of accuracy in the lie detection task.

- We generate synthetic data using the data collected from 42 users to find the correlation between these data and evaluate the performance of our lie detection solution over the synthetic data.

- We train machine learning algorithms using the data collected from 42 users and then enable them in detecting lying over data collected from other five users using the same survey system. We find that we can achieve a good level of accuracy here.

- Subsequently, exploiting findings of the performed tasks, we develop an Android application. The applications can identify lying even with a better level of accuracy after integrating a notion of fault tolerant technique namely NMR.
- Finally, we develop a dynamic end-user application, where one can create an own survey with self-selected questions, collect feedback from different participants, and check lying in the participant's answers. We let 30 users use this system. We find a good level of accuracy with the system.

The rest of the book is organized in the following way. In Chapter 2, we will show the background and related research studies. After that in Chapter 3, we will conduct a survey and analyze the smartphone usage behavior of the participants. In Chapter 6, we discuss the methodology that we use to solve the problem which is formulated in Chapter **??**. In the later chapter, we will show the application development and experimental results in real-life*sec:appDev*. After that, we will have a short conclusion including the future possible research directions.

Chapter 2

Related Work

Research on lie detection has drawn more attention in recent decades [7]. The importance of detecting lies cannot be overlooked in recent times due to its tremendous implication for police interrogation, psychological requirements, judgments of an interviewee, social communication, and other hundreds of real-life essentially. Considering the diversified ways of lie detection, research on lie detection can be categorized into two major groups, social intelligence, and automated lie detection. Another new research is emotion detection using smartphones or computers, however, it can help us in identifying deception.

2.1 Social Intelligence Based Lie Detection

At the very initial stage of research, psychologists started working on lie detection. The major concerned point of the research was to identify behavioral dissimilarity between truths and lying states. Social intelligence such as emotional intelligence [8] and Theory of Mind[9] work with emotion recognition ability through identifying such behavioral dissimilarity. Individuals having high social intelligence are supposed to have a strong capacity to detect emotional cues in faces, which helps to detect lies [10]. In parallel, another similar notion of social intelligence called the theory of mind comes into light. The study in [11] shows that higher scores on a Theory of Mind measure can read cooperators mind more accurately in a Prisoners Dilemma game, and thus can contribute to detection of lies. Implementing such social intelligence, lie detection activities are being conducted in many important cases [12]. To do so, an interviewer with enhanced social intelligence asks several questions to a participant and observes the emotional cues. In this approach, a set of standardized questions is generally used. Here, it is assumed that during the interviews, liars feel more uncomfortable and guilty than truth tellers and display more nervous behaviors such as crossing legs, shifting about in chairs, looking away, etc, [13, 14]. Besides, liars are always unable to provide information as the truth tellers do [15, 16]. If investigators ask more questions liars are less able than truth tellers to cope with these. Liars typically prepare themselves for answering to a set of questions they expect to be asked [17]. Therefore, while exploiting social intelligence, unexpected questions are asked. Truth tellers answer these questions without significant changes in their appearances, however, liars find answering the unanticipated question more difficult than answering the anticipated ones showing significant changes to their appearances [18].

2.2 Bio-Signal Based Automated Lie Detection

automated lie detection technique usually works with a polygraph machine. Here, CIT polygraph test is used when a suspect denies his/her connection to a specific crime. During the test, examinees are given questions with multiple-choice answers (e.g., how did you kill Mr. X? (i) drown her; (ii) strangle her with a rope; (iii) stab her with a knife or (iv) shoot him with a gun?) A deceptive examinee will recognize the correct answer, which produces a (physiological) orienting response, however, truthful suspect does not recognize the correct answer and will not show an orienting response [19]. The polygraph test based on these considerations is a probabilistic test that involves capturing of physiological responses and uses statistical decision theory to calculate the confidence level truth or lie [20, 21, 22, 23, 24]. Here, the major obstacle to make the polygraph test ubiquitous is that it needs specialized machines infrastructure and environment that cannot be deployed everywhere. An intuitive replacement for the machine and infrastructure could be smartphones, which has already gained ubiquitous acceptance, all over the world.

Jang et al., worked on emotion recognition by extracting bio-signal features, i.e., ECG, EDA, PPG and SKT electrodes for the acquisition of bio-signals such as were placed on their bilateral wrists, fingers, and ankle to identify emotions [25]. Zhang et al., presented an innovative emotion recognition approach by extracting the feature of EEG signal and eye tracker under different emotion.

2.3 Smartphone or Computer Based Emotion Detection

In 1997, Picard [7] mentioned the role of emotions in human-computer interaction in the concept of affective computing, which attracted researchers from the different domain, e.g., computer science, psychology, cognitive science, biotechnology, and so on. In the very beginning of emotion detection, researchers started working on text-based emotion identification. [26, 27] used Keyword Spotting, Lexical Affinity Method, Learning-based Methods, and Hybrid Methods for text-based emotion recognition system. Cheng et al., proposed a framework that estimates the sentiment by computing the opinion and lexica extracted from unlabeled data [28, 29]. Extracting the candidate opinion words by the adverb like very and highly, they extract opinions using the lexicon. Finally, the sentiment is computed using a sentiment classifier method. Liu et al., presented an emotion recognition method by extracting textual and non-textual features for micro blogging [30, 31]. Traditional sentiment analysis includes a variety of key words to reflect the emotional state and non-textual features include the common emoticons, temporal features, and punctuation in micro-blog.

With the tremendous increase in the usage of smartphone, emotion recognition using smartphone came into the focus of researchers in recent years. Traditional emotion recognition technologies include facial expression recognition and speech emotion recognition. Khanna et al., presented a method to recognize selected emotion categories from keyboard stroke pattern based on the significant difference between typing speed, the frequency of using backspaces and use of unrelated keys [32]. Jung et al., extracted the features of the change of position and pressure by the sensor wrapped around the mannequin arm, based on which gesture recognition and emotion analysis are conducted [33]. Gao et al., designed a game named Samurai Fruit to capture players touch behavior, and then classified emotion with machine learning algorithms [34]. Gerald et al., proposed a method to detect stress-related changes in the behavior of individuals by using smartphones [35].

None of the previously mentioned works directly for lie detection. Our work initiates a step towards lie detection using the usage patterns collected through smartphone sensors.

Chapter 3

Design of Our Survey System for Data Collection

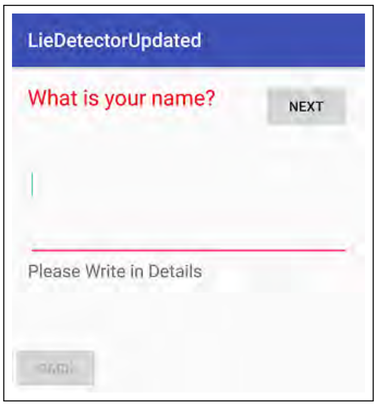
To identify the difference between the smartphone usage behavior while providing true and false feedback, we need to develop an Android application. We use the application to capture responses from participants along with the related usage information. We present the application along with other aspects of our survey system in this chapter.

3.1 Survey Application Development

We develop an Android application (Figure 3.1) for collecting textual data using the questionnaire and present it to the participants. While participating in the survey, we collect the participant's textual feedback including smartphone usage behavior covering typing speed, average shaking, average acceleration, average tilt angle, average rotation, the total number of deleted characters, and the total number of suggestions used, in every millisecond. The reason behind selecting these usage behavior is the availability of sensors in our common smartphones. Usually, we can easily get these values using our normal smartphones and no additional or customized sensors are needed to calculate those values.

In our developed application, we present a single screen to each participant. In the screen, we show different questions one by one. After showing a question, the screen waits for getting a feedback from the participant. For providing the feedback, the participant has to type his/

her answer in response to the displayed question. After submitting each answer, we store all the usage information along with the answer in our server.



What is your name?

Please Write in Details

Figure 3.1: User interface of our application (both survey collection and testing)

Note that, even though our developed application was a simple one to the user, setting up the set of questionnaire used in the application was not a straightforward task. In fact, one of the significant contributions made in our study is the setting up of the set of questionnaire in such a way that we can collect data covering both true and lie answers. We had to judiciously choose the set of questionnaire to do so. We elaborate the process of choosing the set of questionnaire next.

3.2 Selection of Our Set of Questionnaire

One of the major challenges of this research is to identify the set of questionnaire for which participants should provide an informative answer that may be true or false. In another way, we need to make sure that our set of questionnaire contains such questions that exhibit a good possibility of getting false response from a participant.

According to psychology research studies conducted on lie detection, participants usually provide feedback to sensitive questions with false answers [36]. However, further information or guidelines are difficult to find in the literature for preparing survey questions. No sample questionnaire is suggested in the research studies either to the best of our knowledge. To this extent, to get responses from a participant covering both true and lie, we develop a set of questionnaire having several sensitive and non-sensitive questions. We collect the questions

Table 3.1: Some of the survey questions of first failed survey (sequential non-sensitive and sensitive question)

Survey questions	
1. What is your name?	11. What do you like to do?
2. Where do you stay?	12. Tell me about your worst boss/ teacher.
3. What is your date of birth?	13. Did you fall in love?
4. What is your religion?	14. When and how did you start a relationship?
5. Where did you born and grow up?	15. Tell us about your boy/girl friend
6. Tell me a little about yourself.	16. Do you really love him/ her?
7. What is your greatest weakness?	17. Why did you choose him/ her?
8. What are your strengths?	18. When did you make your first sex?
9. What are you most proud of?	19. Do you have any plan to change your boy/ girlfriend?
10. What is your greatest fear?	20. If you were an animal, which one would you want to be?

mostly from different sources [37, 38, 39, 40, 41, 42]. After developing the set of questionnaire, we present them to different participants using our application. Here, we present the sensitive and non-sensitive questions separately - first, we present all the non-sensitive questions one by one, and then we present the sensitive questions one by one. Table 3.1 shows the set of questionnaire as per their orders as presented to the participants.

We invited several participants to participate in our study comprising this first set of questionnaire. Here, we found that the participants provide informative answers only to the non-sensitive questions shown at the beginning. On the other hand, in case of sensitive questions shown in the latter part of the survey, the participants responded with mostly blank answers having invalid answers such as ”Not interested” to a few cases. Thus, our first set of questionnaire failed to collect false responses from the participants. Therefore, we had no other way but to change the set of questionnaire.

Before completely changing the set of questionnaire, an alternative way came to our mind. Here, we plan to explore what would happen if we mix up orders of showing the sensitive and non-sensitive questions to the users. Accordingly, we shuffled their sequences of appearances through showing them in a random order instead of having their fixed sequences as adopted

Table 3.2: Some of the survey questions of second failed survey (mixed and randomized non-sensitive and sensitive question)

Survey questions	
1. What is your name?	11. What do you like to do?
2. Where do you stay?	12. Tell us about your boy/girl friend.
3. What is your date of birth?	13. Why did you choose him/ her?
4. What is your religion?	14. What is your greatest fear?
5. Tell me a little about yourself.	15. Tell me about your worst boss/ teacher.
6. Did you fall in love?	16. What are you most proud of?
7. What is your greatest weakness?	17. Do you really love him/ her?
8. What are your strengths?	18. When did you make your first sex?
9. Where did you born and grow up?	19. Do you have any plan to change your boy/ girlfriend?
10. When and how did you start a relationship?	20. If you were an animal, which one would you want to be?

earlier (non-sensitive questions first and then the sensitive questions). Table 3.2 presents a sample of such sequencing.

Even after randomizing the sequences of sensitive and non-sensitive questions, we failed to get informative answers for the sensitive questions as we did in the earlier case. Here, after collecting responses from several participants with the randomly-ordered set of questionnaire, we found that the participants provide informative answers to only non-sensitive questions retaining the sensitive questions mostly blank or responded with invalid answers. Therefore, now, we are left with no other option but to change the questions.

Accordingly, we designed a new set of questionnaire with mostly new questions comprising both sensitive and non-sensitive questions. Here, we organize the questions in such a way that they request for responses from a participant in a chronological manner staring from the early childhood to current age. We organize the questions in this way for making it easy for a participant to answer the questions in a story-like interview mode. We develop this survey questionnaire with a total of 45 sensitive and non-sensitive questions. Here, we include 17 sensitive questions in different parts of the survey.

We present the new set of questionnaire to several participants. Here, we started to get valid answers in response to both sensitive and non-sensitive questions. The more encouraging facts to us was that, in this case, we started getting false answers from the participants.

Table 3.3: First successful survey questions (question from the participants childhood to current age)

Survey questions

1. What is your name?
2. Where do you stay?
3. Date of birth
4. Religion
5. Where is your hometown?
6. Where did you born and grew up?
7. Who was your best friend in your childhood?
8. Whom did you love most in your childhood?
9. What was your worst memory in school?
10. What is the name of your primary school?
11. How and where was it?
12. Who was your best friend in your high school?
13. Which type of works did you like in school?
14. When and how did you smoke first time?
15. Did you get caught by your parents while smoking? How?
16. What was your favorite movie in school?
17. Did you have any dream girl/boy? Who is s/he?
18. When and did you watch porn first time?
19. What did you after watching porn and how was it?
20. When did you masturbate first time?
21. What did you spent your leisure time in college?
22. Which type of game did you like in college?
23. Did you tease any teacher in college?
24. Did you fall in love of any teacher?
25. When and did you start affair/relationship?
26. What did you with her?
27. Your department and University
28. Why did you choose this subject?
29. What is your future plan?
30. What is the duration of your relationship?
31. Are you virgin?
32. When and how did you make your first kiss?
33. When did you make your sex first time? What was your relationship duration then?
34. Does your family know about your relation?
35. Will you marry your lover? Why?
36. Who is your best friend? Tell us about him/her.
37. Have you ever lied to your best friend?
38. How many times do you watch porn in a week? Which type of porn do you like most?
39. How many times did you masturbate in a week? Dont you feel guilty after doing this?
40. When and why do you tease girls/women in a road or other areas?
41. When was the last time you got laid? With whom?
42. How many partners do you have? Total?
43. What are you plans after your graduation?
44. Why dont you look something better?
45. Why did you answer lie for X (a random number) questions?

Accordingly, we conducted the survey to 42 participants with this new set of questionnaire. We found that most of the participants provide informative answers to both non-sensitive and sensitive questions. Table 3.3 presents a set of questionnaire with which we could able to get the informative answers.

Chapter 4

Conduction of Our Survey and Its Result Analysis

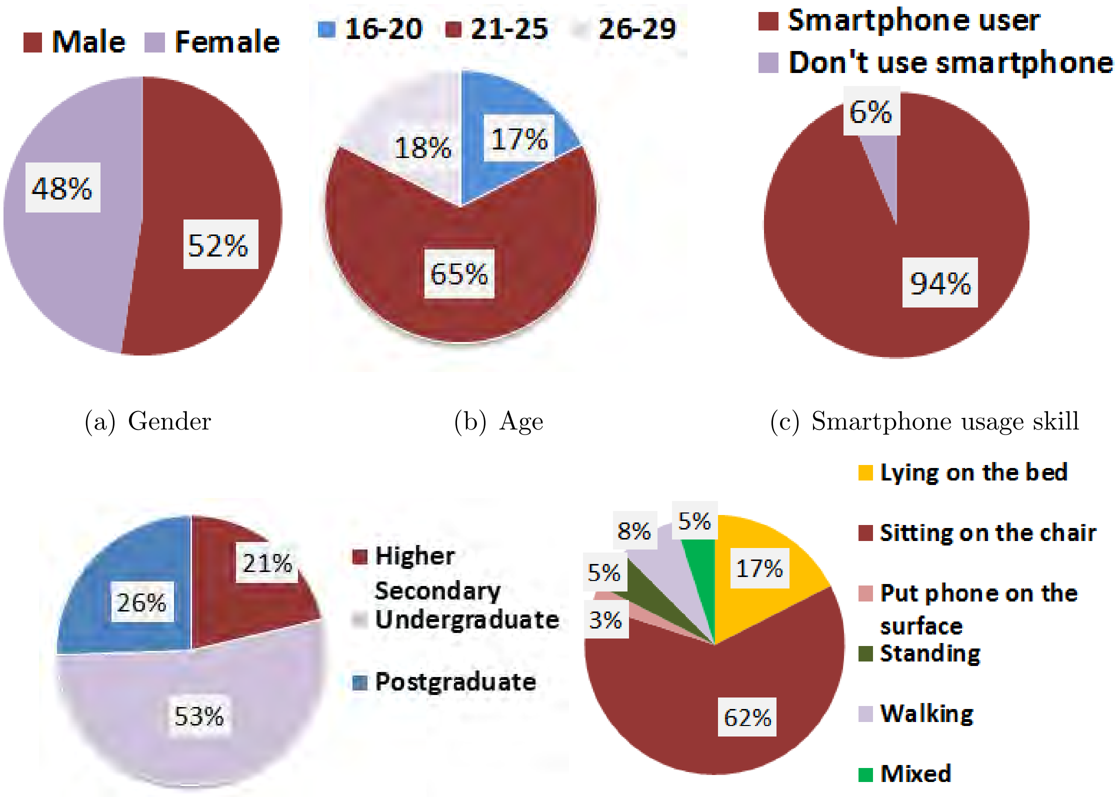
After developing our survey system, we perform step-by-step survey studies over different set of users. In this chapter, we preset our methodology of conducting the survey along with other aspects of the survey.

4.1 Steps and Modes of Conducting the Survey

For getting actual feedback from the survey participants, we invited 42 volunteers. We, initially, briefed on our aim of analyzing usage behavior except for mentioning lie detection. We did this intentionally to get more natural responses. We ensured them on the confidentiality of their answers and requested them to provide true and false answers as per the will of the participants. After that, participants enter a room with our survey system as their own and we ensure that no one can disturb the participants during their participations in the survey. After completing the survey, we provide a hard copy to the participant containing the questions and corresponding answers provided by him/her and conduct a one-to-one interview. We request them to classify his/her answers as TRUE or FALSE, and then to clarify the reason behind their classifications. We maintain the similar process for all 47 participants and get the training dataset for our machine learning algorithms.

4.2 Demography of the Survey Participants

In the initial phase of data collection, we engage 47 participants. 26 of the participants are male and 21 are 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 while sitting on a chair or a bed. Except this, the participants also answer while lying on the bed, Walking in the room, Standing, putting the phone on the table, and having a mix of previous states (5%). Figure 4.1 presents a demography of the participants.



(a) Gender

(b) Age

(c) Smartphone usage skill

(d) Educational qualification

(e) Placement of participant

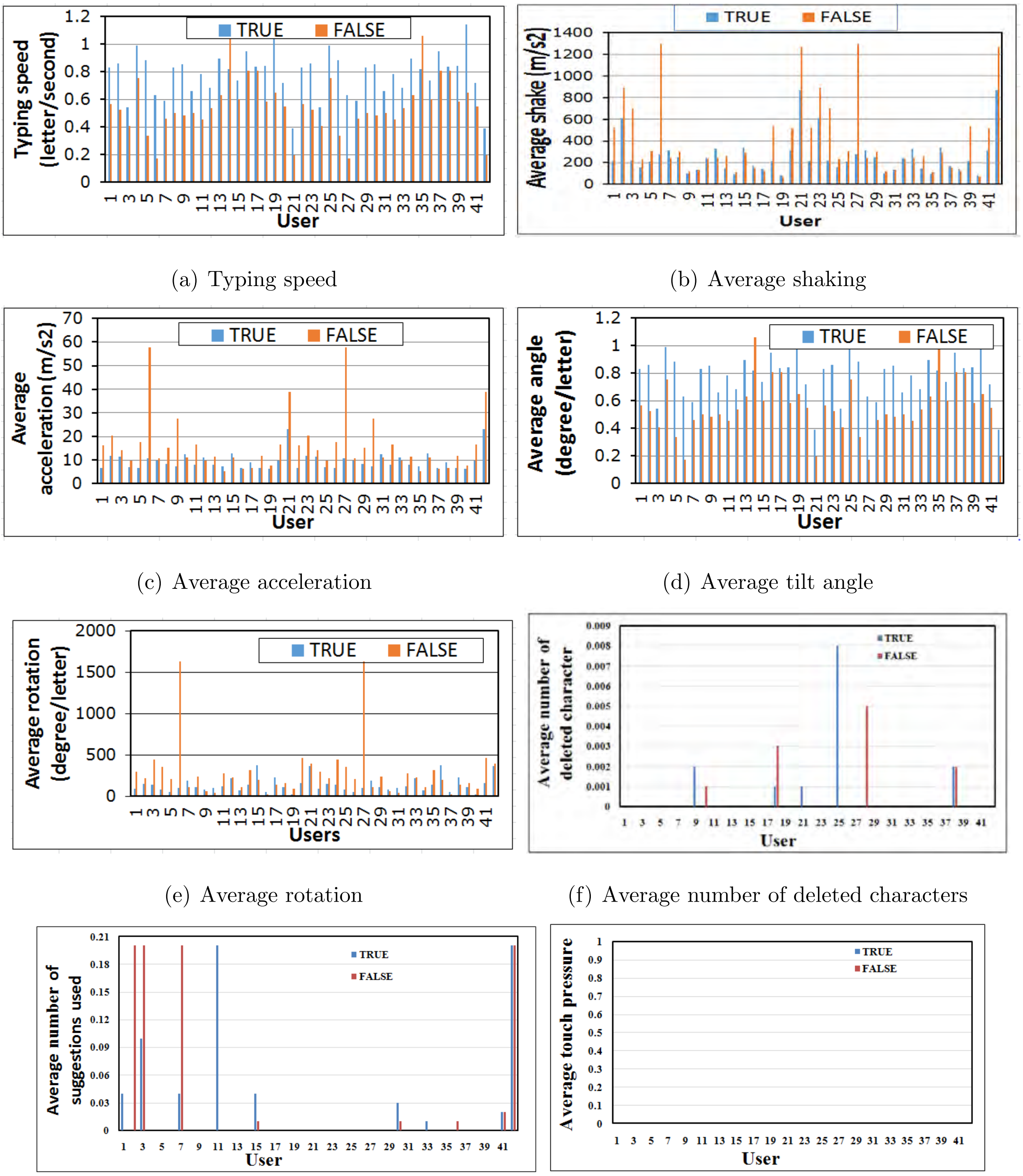
Figure 4.1: Demography of participants

4.3 Feature Extraction

We collected a total 1890 responses from the participants. Subsequently, we plotted average values of eight different features under consideration namely typing speed, average shaking, average acceleration, average tilt angle, average rotation, average number of suggestions used, average number of deleted characters, and average touch pressure. Here, we plot average values of each of the features for both true and false answers side by side. Figure 4.2 presents the plots.

After analyzing the plots, we find that there remain a significant difference in typing speed, average shaking, average acceleration, average tilt angle, and average rotation (Figure 4.2a-4.2e) over the true and false answers for most of the participants. More importantly, the nature of difference remains similar for most of the users. For example, the average typing speed gets lower while providing false answers for most of the participants (having only two exceptions out of 42 participants). However, there is almost no significant difference in average number of suggestions used and average number of deleted characters, as they are not even used by most of the participants. Nonetheless, there is no significant pressure value sensed by the touch sensor while participating in our survey irrespective of the nature of answers being provided.

After analyzing these results, we find that three features namely average number of suggestions used, average number of deleted characters, and average touch pressure cannot be used as features in our solution intended for lie detection. Thus, we adopt the rest five features in our solution for differentiating between true and false answers based on only usage data.



(a) Typing speed

(b) Average shaking

(c) Average acceleration

(d) Average tilt angle

(e) Average rotation

(f) Average number of deleted characters

(g) Average number of suggestions used

(h) Average touch pressure

Figure 4.2: Variation in usages of different participants

4.4 User Evaluation with Machine Learning Algorithms

As 710 answers are False and 1180 answers are True. To resolve this imbalance, we implement Class Balancer algorithm and calculate accuracy using 10 folds cross-validation in five machine learning algorithms using Weka. False acceptance rate (FAR) is the rate of true answers detected as false, whereas, False Rejection Rate (FRR) is the rate of false answers detected as true. The accuracy we found with different machine learning algorithm in Weka is shown in Table 4.1. From the Table 4.1 and 4.2, it is clear that overall performance showed by the Random Forest algorithm is better and can be used for further application development as the core machine learning algorithm.

Table 4.1: Accuracy (%) of several machine learning techniques with our system

Survey questions	Accuracy (%)	Survey questions	Accuracy (%)
BayesNet	77	NaiveBayesMultinomialText	77
Logistic	75	SGDText	75
VotedPerceptron	76	LWL	77
AttributeSelectedClassifier	75	ClassificationViaRegression	73
FilteredClassifier	73	NaiveBayes	77
NaiveBayesUpdatable	77	SGD	75
SimpleLogistic	73	SMO	74
ADABoostM1	75	IterativeClassifierOptimizer	75

Table 4.2: Performance of false detection using usage data using Weka

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

	Algorithms					
Measure	IB1	kStar	RC	RT	RF	Derivations
True Positive Rate or Recall	76%	75%	76%	77%	78%	TPR = TP / (TP + FN)
True Negative Rate	88%	94%	94%	89%	92%	SPC = TN / (FP + TN)
Precision	91%	95%	95%	91%	94%	PPV = TP / (TP + FP)
Negative Predictive Value	72%	68%	69%	72%	73%	NPV = TN / (TN + FN)
False Positive Rate	12%	6%	6%	11%	7%	FPR = FP / (FP + TN)
False Discovery Rate	9%	5%	5%	8%	6%	FDR = FP / (FP + TP)
False Negative Rate	24%	25%	24%	23%	22%	FNR = FN / (FN + TP)
Accuracy	81%	82%	82%	82%	83%	ACC = (TP + TN) / (P + N)
F1 Score	83%	83%	84%	83%	85%	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	64%	66%	67%	65%	68%	TP*TN - FP*FN / sqrt((TP+FP)* (TP+FN)*(TN+FP)*(TN+FN))
False Acceptance Rate	5%	3%	3%	5%	4%	FAR=TN/(TP+TN)
False Rejection Rate	16%	18%	18%	17%	16%	FRR=FP/(FP+FN)

We initially divided our real data set into two sub-parts and synthetic data into another two sub-parts. After that we calculate the accuracy using the best five machine learning algorithms shown in Table 4.2 and implement wilcoxon-signed-rank-test on that accuracy to see whether this difference is statistically significant. Table 4.3 shows the summary of the statistical test and it shows that the null hypothesis has been accepted in favor of Random Forest Algorithm.

Table 4.3: Wilcoxon signed rank test summary

Test	Hypothesis(= 0:05)
IBk	Accepted
kStar	Accepted
Random Tree	Accepted
Random Committee	Accepted

We again collected responses from another five participants same as the previous procedure to check the performance. We trained those six classifiers by a different number of users responses (from 42 users) and then tested on the new five responses. Random Forest shows the highest accuracy first among all classifiers (Figure 3.4).

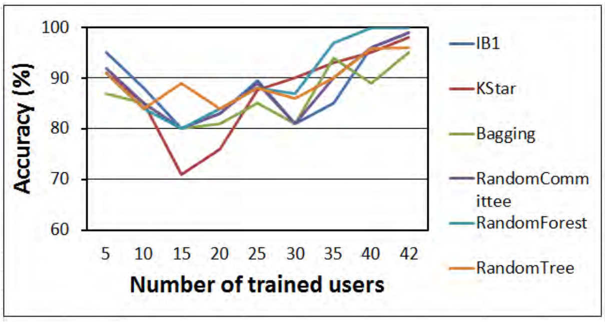


Figure 4.3: Accuracy of lie detection for different number of trained users

4.5 Performance Evaluation of Random Forest Algorithm

As we find an average of 82% accuracy (Table 4.2) on our collected dataset, we develop an application with another set of questions (Table 4.4). The aim of developing this application is to analyze the possibility of detecting lies using the smartphones. This application can predict an answer as true or false in real-time using the initial 47 participants data as the training dataset, Class Balancer, Resample, and Random Forest algorithm. When a false prediction is initiated by the application, the application shows a pop-up window as a warning. The implemented architecture and CPU & memory usage is shown in Figure 4.4 & Figure 4.5 respectively.

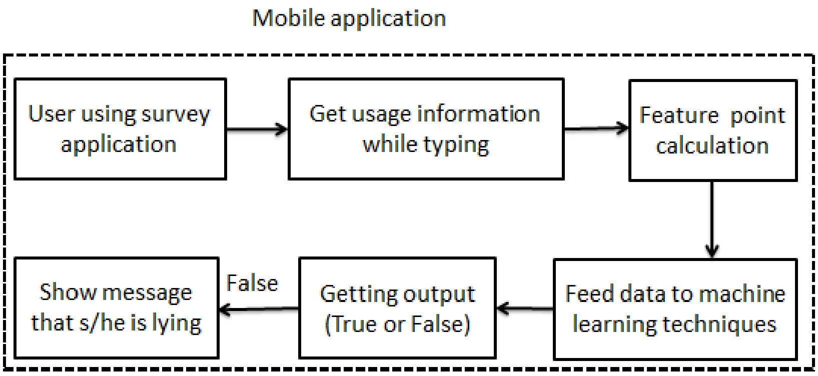

Mobile application
User using survey application
Get usage information while typing
Feature point calculation
Feed data to machine learning techniques
Getting output (True or False)
False
Show message that s/he is lying

Figure 4.4: Mobile application architecture

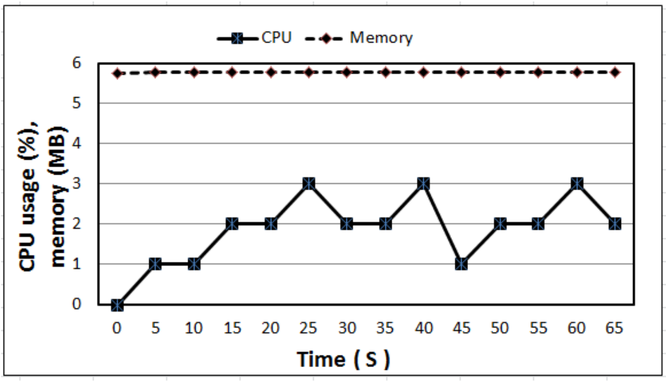


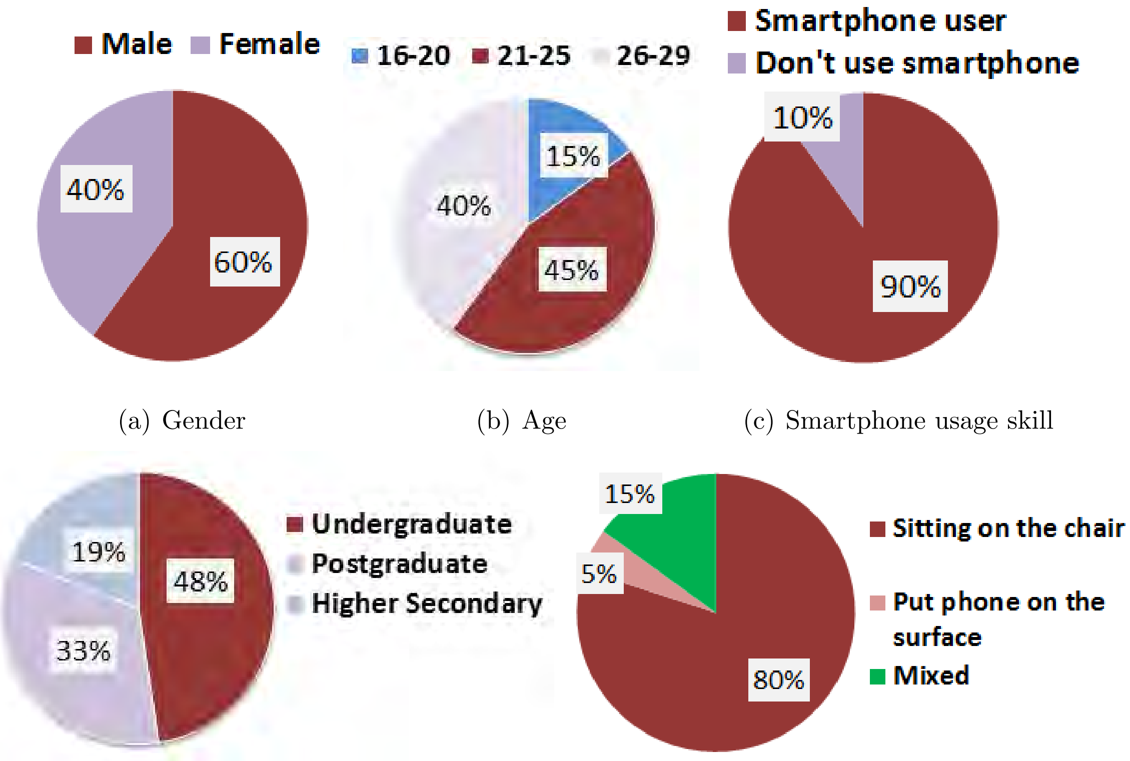
Figure 4.5: CPU and memory usages over time while using the application

Table 4.4: Survey questions used in our lie detector

Survey questions	
1. What is your name?	14. How did you deal the most difficult period in your life?
2. What is your date of birth?	15. Tell me about a time you faced an ethical dilemma.
3. What is your educational status?	16. How do you want to improve yourself in the next year?
4. What is your religion?	17. What are your lifelong dreams?
5. What is your job/occupation?	18. What do you ultimately want to become?
6. What is your greatest weakness?	19. Do you pray regularly? What about your family members?
7. What are your strengths?	20. What is the punishment in your religion for avoiding your prayer?
8. What are you most proud of?	21. What is the opinion about corruption in your country?
9. What is your greatest fear?	22. 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)?
10.What do you like to do?	23. What is the punishment for taking hush money in your religion?
11. Tell me about your worst boss.	24. What is your opinion about usury/interest system in your country?
12. How would you deal with a high-strung personality?	25. Did you ever get usury/interest money?
13. Where do you see yourself in five years?	26. What is the punishment for taking usury/interest money in your religion?

4.5.1 Demography of the Survey Participants

We invited another new 22 volunteers (no overlapping among participants) to participate in our survey using our new survey application. During the survey for performance evaluation, 60% of the participants are male whereas the female is 40%. 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 21-29 years having prior experience on smartphone usage. Besides, most of the participants answer the survey questions while sitting on a chair or a bed. Except this, the participants also answer while lying on the bed, Walking in the room, Standing, putting the phone on the table, and having a mix of previous states. Figure 4.6 presents a demography of the participants.



(a) Gender

(b) Age

(c) Smartphone usage skill

(d) Educational qualification

(e) Placement of participant

Figure 4.6: User demography during performance evaluation

4.5.2 Result of Detecting Lying

We collect 660 answers (433 answers are true and 227 answers are false) from the participants on the result provided by machine learning technique. According to their feedback, we calculate the accuracy of our system and the application returns 96% accuracy (shown in Table 4.5), which indicates the possibility of working on lie detection using smartphones.

Table 4.5: False rate with different classifiers for false detection with Android application while calculating the performance in lie detection using our new set of questions

Measure	Random Forest	Derivations
True Positive Rate or Recall	100%	TPR = TP / (TP + FN)
True Negative Rate	90%	SPC = TN / (FP + TN)
Precision	94%	PPV = TP / (TP + FP)
Negative Predictive Value	100%	NPV = TN / (TN + FN)
False Positive Rate	10%	FPR = FP / (FP + TN)
False Discovery Rate	6%	FDR = FP / (FP + TP)
False Negative Rate	0%	FNR = FN / (FN + TP)
Accuracy	96%	ACC = (TP + TN) / (P + N)
F1 Score	97%	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	92%	TP*TN - FP*FN / sqrt((TP+FP)* (TP+FN)*(TN+FP)*(TN+FN))
False Acceptance Rate	4%	FAR=TN/(TP+TN)
False Rejection Rate	0%	FRR=FP/(FP+FN)

Chapter 5

Synthetic Data Generation

In order to analyze scalability of our proposed solution, we generate synthetic data in much larger scales. Our basis of generating synthetic data is our collected real data. Utilizing the real data, we follow a specific method for generating our synthetic data. Before presenting our approach, we briefly discuss on different techniques available for generating synthetic data.

5.1 Existing Techniques for Synthetic Data Generation

There are several existing techniques used for synthetic data generation. IBM QUEST, GSTD, Mathematical, and Hybrid are some notable techniques for generating synthetic data.

The classical IBM QUEST consists of two data generators [43]. One generator generates training data set and another one generates the testing data set. The generated data sets contain only numerical values and we need to replace or represent textual data using the generated numeric values. QUEST also develops a set of functions for data classification [43]. On the other hand, GSTD uses three parameters namely time, location, and size of the spatiotemporal objects for controlling data generation [44]. Besides, for generating data using the Mathematical technique, we need to identify the distribution of real data set under consideration. After that, we generate synthetic data using the mathematical equation realized from the identified data distribution [45]. Finally, the Hybrid technique initially identifies a cluster where the target data exists, and then implement Mathematical or any other technique for generating synthetic data following the identified cluster [46]. Apart from these techniques, in recent times, machine learning based synthetic data generation techniques have been widely used [47, 48, 49, 50, 51]. Therefore, we first attempt for generating our synthetic data following such techniques using a widely-adopted tool namely Auto Weka.

5.2 Synthetic Data Generation using Auto-Weka

We use Auto-Weka [52], a feature of Weka [53], for our synthetic adata generation purpose. Auto-Weka is a tool that helps to identify best machine learning algorithm for a given dataset. It helps to predict the machine learning algorithm and its attributes to detect an existing pattern inside the dataset under consideration. Exploiting this feature, we apply several algorithms to our real dataset and attempts to obtain existence of a pattern in our data as revealed by Auto-Weka. For this purpose, we experiment on identifying relationship between demography and real user data. In order to do so, we calculate averages and standard deviations of five normalized features for both "True" and "False" labels. Thus, we generate ten columns for all real-user data. Later, we apply Auto-Weka to predict the best machine learning algorithms according to relationship between demography and usage data. However, Auto-Weka fails to identify any machine learning algorithm in six cases out the ten cases. Thus, we conclude that synthetic data generation in our case is not possible based on the user demography data.

5.3 Synthetic Data Generation using Mathematical Technique

Next, we attempt to generate our synthetic data following the Mathematical technique. To do so, first, we identify that there might be normality in our real data, as they represent real usage data. To confirm it, we perform normality testing over the real data.

5.3.1 Normality Testing Using Shapiro Wilk Test

To check whether the real data collected from the participants are normally distributed or not, we use Shapiro Wilk test in R. Here, we check whether the mean values of typing speed, acceleration, shaking, angle, and rotation are normally distributed or not. Accordingly, we perform Shapiro Wilk test and present the results in Table 5.1. The results demonstrate high potential of possessing normality in the real data.

Table 5.1: Results of Shapiro Wilk test on our real dataset

Feature	W	P-value
Mean (typing speed)	0.99911	0.9146
Mean (average shaking)	0.92845	0.9146
Mean (average acceleration)	0.99782	0.194
Mean (average angle)	0.99234	0.9146
Mean (average rotation)	0.96261	0.194

5.3.2 Chi-Square Goodness-of-Fit Test

We also perform Chi-square goodness-of-fit test over our real data for determining inherent probability distribution of the data and parameters of the distribution determined from the data. Here, we consider a null hypothesis that our real data follows a Normal probability distribution. Next, we apply chi^2gof(x) function from Matlab to our normalized features, where x represents a vector. chi^2gof(x) returns a test decision for the null hypothesis that the data in vector x comes from a Normal distribution with a mean and variance estimated from x, using the Chi-square goodness-of-fit test. If the returned result is 0, then we can establish that our specific feature follows Normal distribution at 5% significance level. On the other hand, if the result is 0, we can say that null hypothesis is rejected, which means that specific feature does not follow Normal distribution.

Table 5.2: Chi-Square Goodness-of-Fit test for all four normalized features (Hypothesis result = 0 means null hypothesis is accepted and data follows Normal distribution)

Normalized feature	Hypothesis result
Typing speed	0
Average shaking	0
Average acceleration	0
Average angle	0
Average rotation	0

We apply the function to all normalized features in our data and find zero for all the normalized features as presented in Table 5.2. Thus, we can conclude that our real data follows Normal distribution, which makes it eligible to apply in fitdist for generating synthetic data from the statistical model Normal distribution.

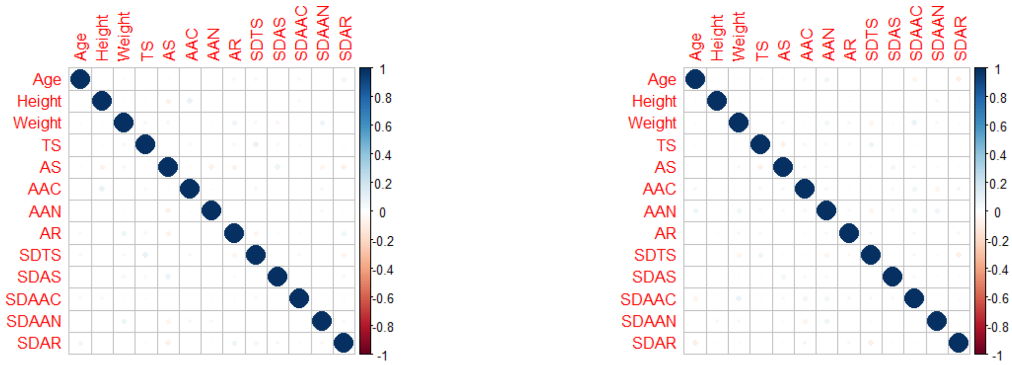
In addition to identifying the probability distribution, it is of utmost significance to also identify whether our considered features are independent or correlated. To identify the correlation, we perform Pearson's test.

5.3.3 Correlation Analysis Using Pearson's Algorithm

Correlation is a measurable strategy that can indicate whether and how strongly variables are connected to each other. The estimation of this connection coefficient differs over the range from +1 to -1. An estimation of 1 demonstrates an ideal level of relationship between the two variables. As the connection coefficient goes towards 0, the connection between the two variables becomes weaker. A + sign demonstrates a positive relationship and a sign demonstrates a negative relationship.

Pearson r correlation is the most broadly utilized relationship measurement of the level of the connection between directly related variables. Therefore, we adopt it in our analysis on revealing correlation among the features under consideration. Further, to validate the already-found impact of demographic attributes, we include the demographic attributes in our correlation test.

Figures 5.1 presents results found from our features (both of their averages and standard deviations) along with the demographic attributes. The figure demonstrates that the demographic attributes do not exhibit any correlation (or impact) on the features under consideration. Besides, the features (both of their averages and standard deviations) under consideration do not exhibit any correlation to each other. Therefore, the features are worth of adopting for synthetic data generation. We perform the synthetic data generation using Matlab.



(a) Correlation for true data

(b) Correlation for false data

Figure 5.1: Correlation between different attributes for true and false answers

5.4 Data Generation Using MatLab

As we have found the Normal distribution over our real data, we generate 2000 users synthetic data exploiting the Normal distribution using the following equation:

$$R = normrnd(\mu, \sigma) \tag{5.1}$$

This equation generates random numbers following the Normal distribution with mean parameter μ and standard deviation parameter σ.

5.4.1 Validation of Our Generated Synthetic Data

To validate consistency of our generated synthetic data, we initially generate synthetic data of 42 users with the help of the real data collected from the participants. We can validate the generated data in case we find the following -

- 10-fold cross-validation accuracy for real and synthetic data are similar.
- Training and testing over the real data and synthetic data in both directions result in similar accuracy.

Accordingly, we test the real and synthetic data using Random Forest algorithm and 10 fold cross validation technique in Weka. Here, we find 84% accuracy for real data and 88% accuracy for synthetic data (Table 5.3), which is close to each other. Besides, we perform testing over the synthetic data after using the real data as training data and testing over the real data after using the synthetic data as training data. Here, we get an exactly same accuracy of 94% accuracy in both the cases (Table 5.4). Therefore, we can conclude that the generated synthetic data is consistent with respect to the real data.

Table 5.3: False rate with Random Forest for false detection with 42 users real and synthetic data suing 10 fold cross-validation

Measure	Real Data	Synthetic Data
True Positive Rate or Recall	82%	83%
True Negative Rate	86%	93%
Precision	88%	94%
Negative Predictive Value	80%	81%
False Positive Rate	14%	7%
False Discovery Rate	13%	6%
False Negative Rate	18%	16%
Accuracy	84%	88%
F1 Score	85%	88%
Matthews Correlation Coefficient	68%	76%
False Acceptance Rate	6%	0%
False Rejection Rate	6%	13%

Table 5.4: False rate with Random Forest for false detection with 42 users real and synthetic data

Measure	Train: Real Data Test: Synthetic Data	Train: Synthetic Data Test: Real Data
True Positive Rate or Recall	94%	89%
True Negative Rate	94%	100%
Precision	94%	100%
Negative Predictive Value	94%	87%
False Positive Rate	6%	0%
False Discovery Rate	6%	0%
False Negative Rate	6%	11%
Accuracy	94%	94%
F1 Score	94%	94%
Matthews Correlation Coefficient	88%	88%
False Acceptance Rate	6%	0%
False Rejection Rate	6%	13%

5.4.2 Data Training and Testing using Weka

We generate synthetic data for 2000 users. For each user, we generate two different of data (one for true data and another for false data) resulting in a total of 4000 data. This 4000 data makes our synthetic dataset. We train Random Forest algorithm with synthetics data of different number of users and test on the data of remaining users. Accordingly, we get different results. Table 5.5 to 5.15 present the different results.

5.4.3 Detection Results over Synthetic Data

We calculate and analyze the performances pertinent to different numbers of training and testing data using Random Forest algorithm. Figure 5.2 presents the results. It is evident from the graph that the accuracy increases sharply up to the size of training dataset to be 100. After that, accuracy increases marginally, as the accuracy increases by up to only 4% through increasing the size of training dataset up to 900. This result demonstrates that training our

Table 5.5: False rate with Random Forest for false detection with 30 users training data

Measure	RF	Derivations
True Positive Rate or Recall	68%	TPR = TP / (TP + FN)
True Negative Rate	61%	SPC = TN / (FP + TN)
Precision	60%	PPV = TP / (TP + FP)
Negative Predictive Value	69%	NPV = TN / (TN + FN)
False Positive Rate	39%	FPR = FP / (FP + TN)
False Discovery Rate	40%	FDR = FP / (FP + TP)
False Negative Rate	32%	FNR = FN / (FN + TP)
Accuracy	64%	ACC = (TP + TN) / (P + N)
F1 Score	64%	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	29%	TP*TN - FP*FN / sqrt((TP+FP)* (TP+FN)*(TN+FP)*(TN+FN))
False Acceptance Rate	40%	FAR=TN/(TP+TN)
False Rejection Rate	31%	FRR=FP/(FP+FN)

Table 5.6: False rate with Random Forest for false detection with 50 users training data

Measure	RF	Derivations
True Positive Rate or Recall	71%	TPR = TP / (TP + FN)
True Negative Rate	77%	SPC = TN / (FP + TN)
Precision	80%	PPV = TP / (TP + FP)
Negative Predictive Value	68%	NPV = TN / (TN + FN)
False Positive Rate	23%	FPR = FP / (FP + TN)
False Discovery Rate	20%	FDR = FP / (FP + TP)
False Negative Rate	29%	FNR = FN / (FN + TP)
Accuracy	74%	ACC = (TP + TN) / (P + N)
F1 Score	75%	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	48%	TP*TN - FP*FN / sqrt((TP+FP)* (TP+FN)*(TN+FP)*(TN+FN))
False Acceptance Rate	20%	FAR=TN/(TP+TN)
False Rejection Rate	32%	FRR=FP/(FP+FN)

Table 5.7: False rate with Random Forest for false detection with 70 users training data

Measure	RF	Derivations
True Positive Rate or Recall	80%	TPR = TP / (TP + FN)
True Negative Rate	84%	SPC = TN / (FP + TN)
Precision	85%	PPV = TP / (TP + FP)
Negative Predictive Value	79%	NPV = TN / (TN + FN)
False Positive Rate	16%	FPR = FP / (FP + TN)
False Discovery Rate	15%	FDR = FP / (FP + TP)
False Negative Rate	20%	FNR = FN / (FN + TP)
Accuracy	82%	ACC = (TP + TN) / (P + N)
F1 Score	83%	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	64%	TP*TN - FP*FN / sqrt((TP+FP)* (TP+FN)*(TN+FP)*(TN+FN))
False Acceptance Rate	16%	FAR=TN/(TP+TN)
False Rejection Rate	21%	FRR=FP/(FP+FN)

Table 5.8: False rate with Random Forest for false detection with 100 users training data

Measure	RF	Derivations
True Positive Rate or Recall	87%	TPR = TP / (TP + FN)
True Negative Rate	85%	SPC = TN / (FP + TN)
Precision	84%	PPV = TP / (TP + FP)
Negative Predictive Value	87%	NPV = TN / (TN + FN)
False Positive Rate	15%	FPR = FP / (FP + TN)
False Discovery Rate	16%	FDR = FP / (FP + TP)
False Negative Rate	13%	FNR = FN / (FN + TP)
Accuracy	86%	ACC = (TP + TN) / (P + N)
F1 Score	86%	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	71%	TP*TN - FP*FN / sqrt((TP+FP)* (TP+FN)*(TN+FP)*(TN+FN))
False Acceptance Rate	16%	FAR=TN/(TP+TN)
False Rejection Rate	13%	FRR=FP/(FP+FN)

Table 5.9: False rate with Random Forest for false detection with 200 users training data

Measure	RF	Derivations
True Positive Rate or Recall	90%	TPR = TP / (TP + FN)
True Negative Rate	87%	SPC = TN / (FP + TN)
Precision	86%	PPV = TP / (TP + FP)
Negative Predictive Value	90%	NPV = TN / (TN + FN)
False Positive Rate	13%	FPR = FP / (FP + TN)
False Discovery Rate	14%	FDR = FP / (FP + TP)
False Negative Rate	10%	FNR = FN / (FN + TP)
Accuracy	88%	ACC = (TP + TN) / (P + N)
F1 Score	88%	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	76%	TP*TN - FP*FN / sqrt((TP+FP)* (TP+FN)*(TN+FP)*(TN+FN))
False Acceptance Rate	14%	FAR=TN/(TP+TN)
False Rejection Rate	10%	FRR=FP/(FP+FN)

Table 5.10: False rate with Random Forest for false detection with 300 users training data

Measure	RF	Derivations
True Positive Rate or Recall	90%	TPR = TP / (TP + FN)
True Negative Rate	87%	SPC = TN / (FP + TN)
Precision	86%	PPV = TP / (TP + FP)
Negative Predictive Value	90%	NPV = TN / (TN + FN)
False Positive Rate	13%	FPR = FP / (FP + TN)
False Discovery Rate	14%	FDR = FP / (FP + TP)
False Negative Rate	10%	FNR = FN / (FN + TP)
Accuracy	88%	ACC = (TP + TN) / (P + N)
F1 Score	88%	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	76%	TP*TN - FP*FN / sqrt((TP+FP)* (TP+FN)*(TN+FP)*(TN+FN))
False Acceptance Rate	14%	FAR=TN/(TP+TN)
False Rejection Rate	10%	FRR=FP/(FP+FN)

Table 5.11: False rate with Random Forest for false detection with 400 users training data

Measure	RF	Derivations
True Positive Rate or Recall	90%	TPR = TP / (TP + FN)
True Negative Rate	88%	SPC = TN / (FP + TN)
Precision	88%	PPV = TP / (TP + FP)
Negative Predictive Value	90%	NPV = TN / (TN + FN)
False Positive Rate	12%	FPR = FP / (FP + TN)
False Discovery Rate	12%	FDR = FP / (FP + TP)
False Negative Rate	10%	FNR = FN / (FN + TP)
Accuracy	89%	ACC = (TP + TN) / (P + N)
F1 Score	89%	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	78%	TP*TN - FP*FN / sqrt((TP+FP)* (TP+FN)*(TN+FP)*(TN+FN))
False Acceptance Rate	12%	FAR=TN/(TP+TN)
False Rejection Rate	10%	FRR=FP/(FP+FN)

Table 5.12: False rate with Random Forest for false detection with 500 users training data

Measure	RF	Derivations
True Positive Rate or Recall	90%	TPR = TP / (TP + FN)
True Negative Rate	88%	SPC = TN / (FP + TN)
Precision	88%	PPV = TP / (TP + FP)
Negative Predictive Value	90%	NPV = TN / (TN + FN)
False Positive Rate	12%	FPR = FP / (FP + TN)
False Discovery Rate	12%	FDR = FP / (FP + TP)
False Negative Rate	10%	FNR = FN / (FN + TP)
Accuracy	89%	ACC = (TP + TN) / (P + N)
F1 Score	89%	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	78%	TP*TN - FP*FN / sqrt((TP+FP)* (TP+FN)*(TN+FP)*(TN+FN))
False Acceptance Rate	12%	FAR=TN/(TP+TN)
False Rejection Rate	10%	FRR=FP/(FP+FN)

Table 5.13: False rate with Random Forest for false detection with 600 users training data

Measure	RF	Derivations
True Positive Rate or Recall	90%	TPR = TP / (TP + FN)
True Negative Rate	90%	SPC = TN / (FP + TN)
Precision	90%	PPV = TP / (TP + FP)
Negative Predictive Value	90%	NPV = TN / (TN + FN)
False Positive Rate	10%	FPR = FP / (FP + TN)
False Discovery Rate	10%	FDR = FP / (FP + TP)
False Negative Rate	10%	FNR = FN / (FN + TP)
Accuracy	90%	ACC = (TP + TN) / (P + N)
F1 Score	90%	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	80%	TP*TN - FP*FN / sqrt((TP+FP)* (TP+FN)*(TN+FP)*(TN+FN))
False Acceptance Rate	10%	FAR=TN/(TP+TN)
False Rejection Rate	10%	FRR=FP/(FP+FN)

Table 5.14: False rate with Random Forest for false detection with 700 users training data

Measure	RF	Derivations
True Positive Rate or Recall	90%	TPR = TP / (TP + FN)
True Negative Rate	90%	SPC = TN / (FP + TN)
Precision	90%	PPV = TP / (TP + FP)
Negative Predictive Value	90%	NPV = TN / (TN + FN)
False Positive Rate	10%	FPR = FP / (FP + TN)
False Discovery Rate	10%	FDR = FP / (FP + TP)
False Negative Rate	10%	FNR = FN / (FN + TP)
Accuracy	90%	ACC = (TP + TN) / (P + N)
F1 Score	90%	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	80%	TP*TN - FP*FN / sqrt((TP+FP)* (TP+FN)*(TN+FP)*(TN+FN))
False Acceptance Rate	10%	FAR=TN/(TP+TN)
False Rejection Rate	10%	FRR=FP/(FP+FN)

Table 5.15: False rate with Random Forest for false detection with 800 users training data

Measure	RF	Derivations
True Positive Rate or Recall	89%	TPR = TP / (TP + FN)
True Negative Rate	93%	SPC = TN / (FP + TN)
Precision	93%	PPV = TP / (TP + FP)
Negative Predictive Value	89%	NPV = TN / (TN + FN)
False Positive Rate	7%	FPR = FP / (FP + TN)
False Discovery Rate	7%	FDR = FP / (FP + TP)
False Negative Rate	10%	FNR = FN / (FN + TP)
Accuracy	91%	ACC = (TP + TN) / (P + N)
F1 Score	91%	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	82%	TP*TN - FP*FN / sqrt((TP+FP)* (TP+FN)*(TN+FP)*(TN+FN))
False Acceptance Rate	7%	FAR=TN/(TP+TN)
False Rejection Rate	11%	FRR=FP/(FP+FN)

Table 5.16: False rate with Random Forest for false detection with 900 users training data

Measure	RF	Derivations
True Positive Rate or Recall	89%	TPR = TP / (TP + FN)
True Negative Rate	93%	SPC = TN / (FP + TN)
Precision	93%	PPV = TP / (TP + FP)
Negative Predictive Value	89%	NPV = TN / (TN + FN)
False Positive Rate	7%	FPR = FP / (FP + TN)
False Discovery Rate	7%	FDR = FP / (FP + TP)
False Negative Rate	10%	FNR = FN / (FN + TP)
Accuracy	91%	ACC = (TP + TN) / (P + N)
F1 Score	91%	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	82%	TP*TN - FP*FN / sqrt((TP+FP)* (TP+FN)*(TN+FP)*(TN+FN))
False Acceptance Rate	7%	FAR=TN/(TP+TN)
False Rejection Rate	11%	FRR=FP/(FP+FN)

solution with ∼100 users could suffice as per our real data. Moreover, the results pertinent to accuracy obtained through testing over a large number users in the synthetic data confirms scalability of our proposed solution.

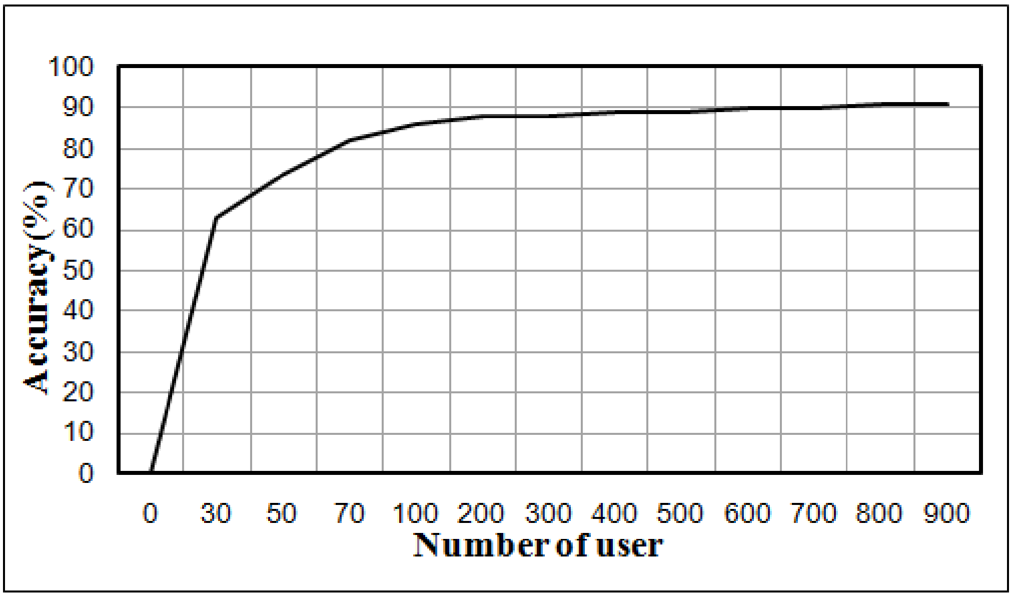


Figure 5.2: Accuracy of lie detection for different number of trained users

Chapter 6

Proposed Architecture

From the Chapter 3, it is clear that there is a significant difference in smartphone usage behavior while providing true and false feedback. Therefore, we start analyzing the possible architectures of the ubiquitous solution for lie detection. Figure 6.1 demonstrates the proposed architecture for our new lie detection system using a smartphone.

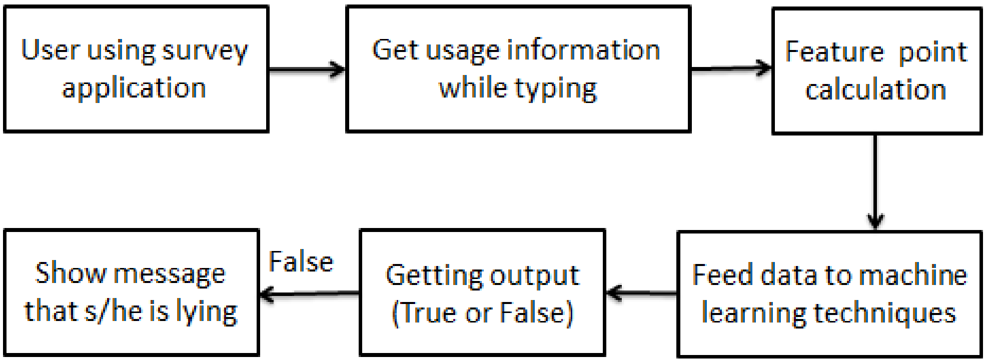

User using survey application
Get usage information while typing
Feature point calculation
Feed data to machine learning techniques
Getting output (True or False)
False
Show message that s/he is lying

Figure 6.1: Accuracy of lie detection for different number of trained users

Our Proposed architecture works as follows:

i. In the beginning, a user starts using our android application or another Android application having the feature or plugin of our technique.

ii. Our application always collects the smartphone users usage behavior and stores it.

iii. Our system analyzes the smartphone usage behavior and feeds it to the machine learning algorithms.

iv. If our system finds the behavior as false then it shows a warning message.

Our system can be developed in two features, i.e., Android application with the complete feature, and Android application with machine learning algorithm on the server.

6.1 Android application with complete feature

In this type of application all processes of data collection, analysis, and detection are done in the smartphone.

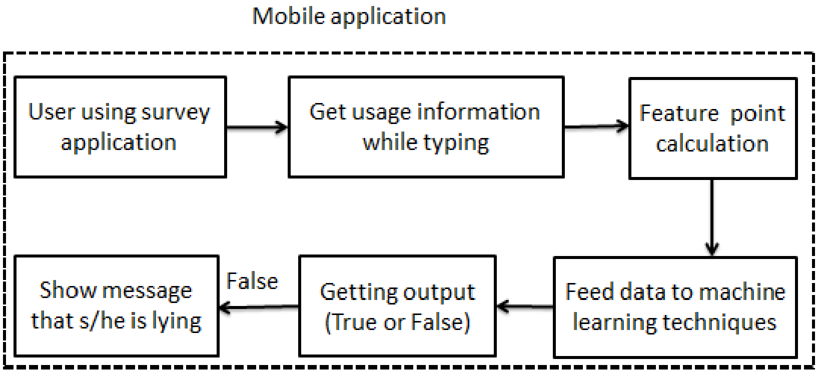

Mobile application
User using survey application
Get usage information while typing
Feature point calculation
Feed data to machine learning techniques
Getting output (True or False)
False
Show message that s/he is lying

Figure 6.2: Accuracy of lie detection for different number of trained users

6.2 Android application with machine learning algorithm in server

In this type of application data collection is done by the smartphone and then it is stored it on the server. The server analyzes the data and works for lie detection. If the result of the analysis becomes false, it sends a false message to the smartphone and smartphone shows a warning in real time.

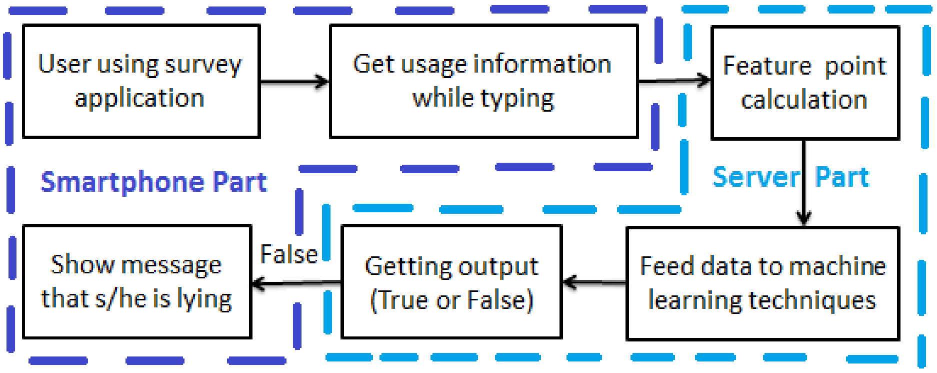

User using survey application
Get usage information while typing
Feature point calculation
Smartphone Part
Server Part
Show message that s/he is lying
False
Getting output (True or False)
Feed data to machine learning techniques

Figure 6.3: Accuracy of lie detection for different number of trained users

Chapter 7

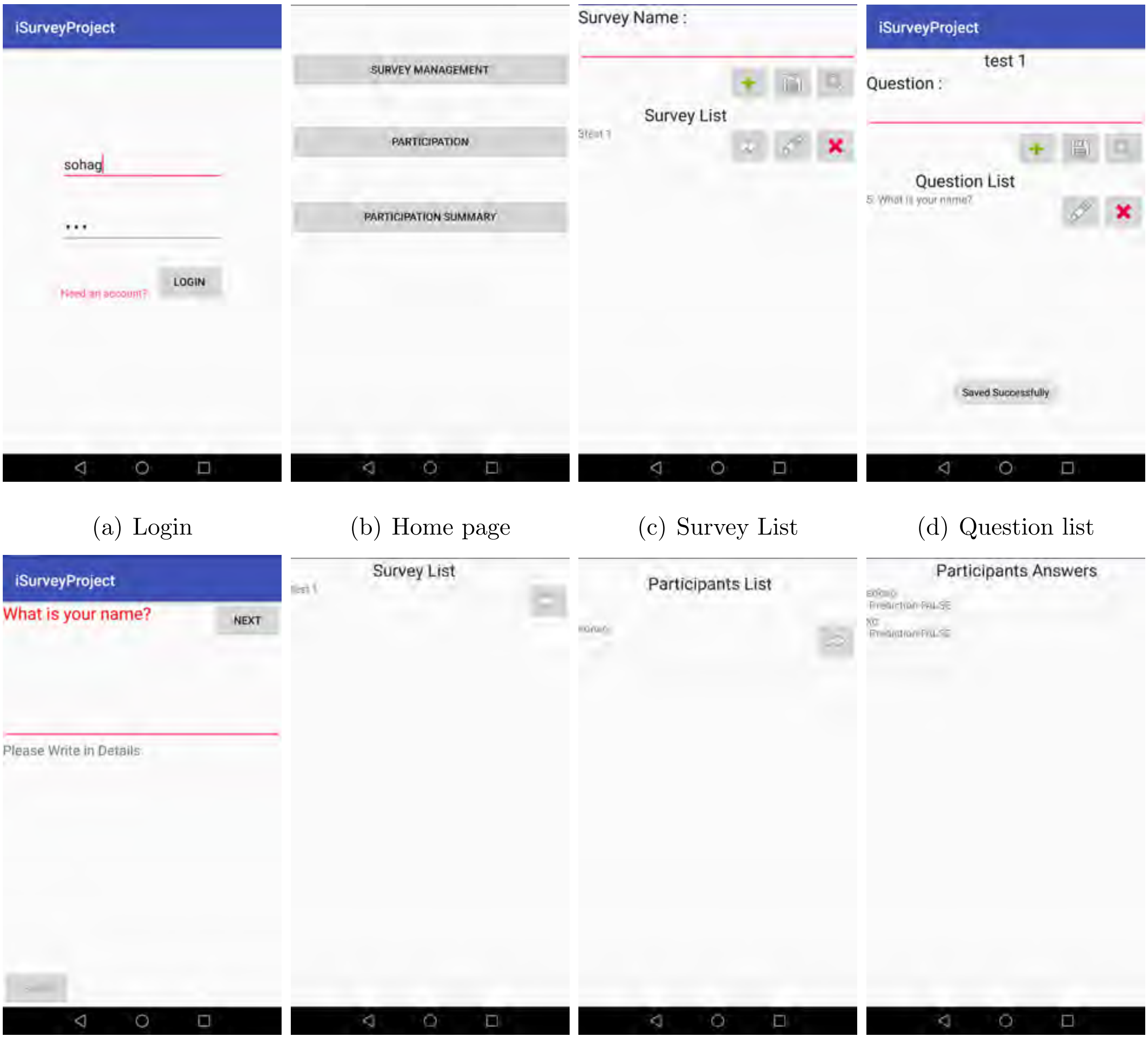
Application Development Based on Our Proposed Architecture

We plan to develop an independent application consisting our solution for detecting lie. To do so, we adopt our proposed architecture presented in the last chapter.

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 himself 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. In this chapter, we present development of the dynamic application along with our technique adopted to enhance accuracy of the developed application further.

7.1 Dynamic Application Development and Its Performance Evaluation

We develop an Android application where a user can create his/ her account and then access the system after login (Fig 7.1a). Three options are shown in the home page therein (Fig 7.1b), where the user can manage survey, conduct the survey, and check the summaries. In our survey management, the user can create surveys (Fig 7.1c) and then add questions (Fig 7.1d), which will be 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 his/her feedback (Fig 7.1e). 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 (Fig. 7.1f - 7.1h). Using our developed application, a user is free to conduct multiple surveys with own-selected questions.



(a) Login

(b) Home page

(c) Survey List

(d) Question list

(e) Answer submission page

(f) Survey list in summary

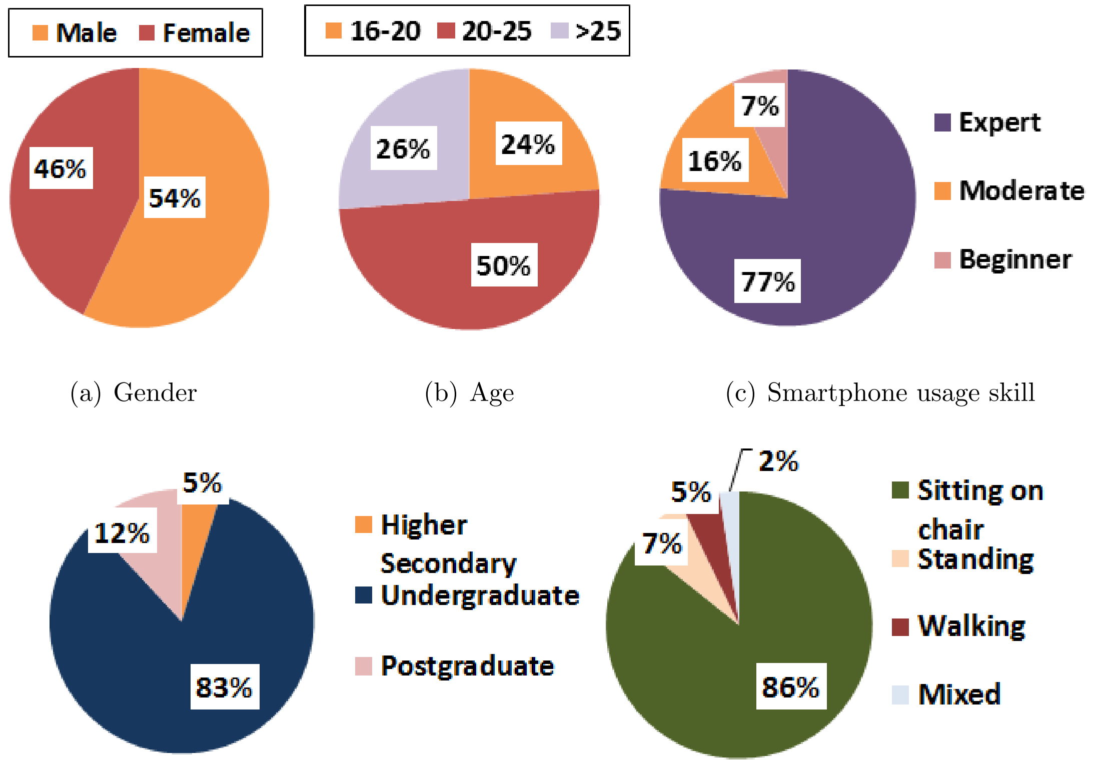
(g) Participants list in summary

(h) Detection summary for individual participant

Figure 7.1: Architecture of dynamic application

7.1.1 Demography of the Survey Participants

We provided our application to new 30 participants (no overlapping among participants) recruited voluntarily. They collected feedback 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 while sitting on a chair or a bed. Except this, the participants also answer while lying on the bed, Walking in the room, Standing, putting the phone on the table, and having a mix of previous states. Figure 7.2 presents a demography of the participants.



(a) Gender

(b) Age

(c) Smartphone usage skill

(d) Educational qualification

(e) Placement of participant

Figure 7.2: User demography during performance evaluation of dynamic application

7.1.2 Evaluation Results

We finally collect feedback from the 30 volunteers on our application. The 42 participants provide their 420 answers (295 is True and 145 is False) as their own style, gossiping with others, etc. They maintain noising environment during the survey and found an average of 84% accuracy (shown in Table 7.1). As our training dataset is not too big, this performance still provides hope for working with lie detection using smartphones.

Table 7.1: False rate with different classifiers for false detection with dynamic end user Android application. This result is collected from the real life users conducting their survey using our application and their self-directed questions

Measure	Random Forest	Derivations
True Positive Rate or Recall	91%	TPR = TP / (TP + FN)
True Negative Rate	74%	SPC = TN / (FP + TN)
Precision	86%	PPV = TP / (TP + FP)
Negative Predictive Value	83%	NPV = TN / (TN + FN)
False Positive Rate	26%	FPR = FP / (FP + TN)
False Discovery Rate	14%	FDR = FP / (FP + TP)
False Negative Rate	9%	FNR = FN / (FN + TP)
Accuracy	84%	ACC = (TP + TN) / (P + N)
F1 Score	88%	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	67%	TP*TN - FP*FN / sqrt((TP+FP)* (TP+FN)*(TN+FP)*(TN+FN))
False Acceptance Rate	10%	FAR=TN/(TP+TN)
False Rejection Rate	6%	FRR=FP/(FP+FN)

7.2 Improving Performance using NMR

NMR stands for N-Modular Redundancy, in which three systems perform a process and that result is processed by a majority-voting system to produce a single output. If any one of the N systems fails, the other N-1 systems can correct and mask the fault. In this state of our research, we implement NMR to improve the performance of our dynamic end-user application.

Therefore, we test the different combination of selected five machine learning algorithms, i.e., IB1, kStar, Random Committee, Random Forest, and Random Tree. The combinations are as follows:

1. At least three algorithms among five algorithms detect a lie.
2. At least two algorithms among IB1, kStar, and Random Forest detect a lie.
3. At least two algorithms among Random Committee, Random Forest, and Random Tree detect a lie.
4. At least one algorithms among IB1, kStar, and Random Forest detect a lie.

Considering the following conditions, we select different the final prediction as false. Based on this decision we calculate the accuracy of different combination using the initially collected 42 participants dataset. The accuracy is shown in Table 7.2. Here, it is clear that NMR based combination improves the performance when N=3 and the algorithms are IB1, kStar, and Random Forest detect a lie. Therefore, we decide to develop our application using the combination type 3.

Table 7.2: Accuracy of different combination of machine learning algorithms

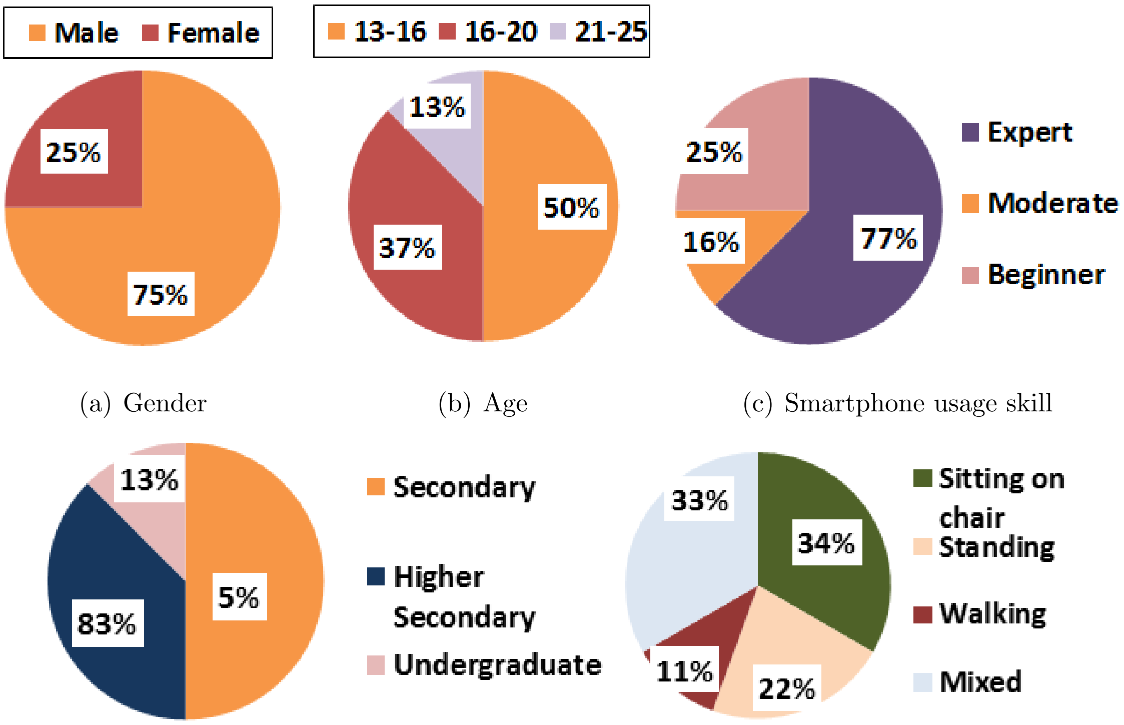
Combination Type	Accuracy
(1)	78%
(2)	88%
(3)	80%
(4)	58%

7.2.1 NMR Based Application Development

In this state of research, we develop another application using combination type 3. This application contains all the features of the previous version (Figure 7.2). The main difference between these versions is the methodology of detecting lie. In the previous version we use only Random Forest algorithm for lie detection, however, we are using NMR technique of the combination type 3. Finally we invited another 8 participants to use our system.

7.2.2 Demography of the Survey Participants

To justify the accuracy, we engage new 8 participants (no overlapping among participants) to conduct their survey using our application. During the survey for performance evaluation, 60% of the participants are male whereas the female is 40%. 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 12-15 years having prior experience on smartphone usage. Besides, most of the participants answer the survey questions while sitting on a chair or a bed. Except this, the participants also answer while lying on the bed, Walking in the room, Standing, putting the phone on the table, and having a mix of previous states. Figure 7.3 presents a demography of the participants.



(a) Gender (b) Age (c) Smartphone usage skill

(d) Educational qualification (e) Placement of participant

Figure 7.3: User demography during performance evaluation of NMR based application

7.2.3 Result

We finally collect feedback from the 10 volunteers on our application. The participants provide 100 answers (60 is True and 40 is False) in their own style, gossiping with others, etc, They maintain noising environment during the survey and found an average of 94% accuracy (shown in Table 7.3). This performance indicates the possibility of smartphone usage as an effective lie detector.

Table 7.3: False rate with different classifiers for false detection with dynamic end user Android application using NMR

Measure	NMR	Derivations
True Positive Rate or Recall	97%	TPR = TP / (TP + FN)
True Negative Rate	90%	SPC = TN / (FP + TN)
Precision	93%	PPV = TP / (TP + FP)
Negative Predictive Value	95%	NPV = TN / (TN + FN)
False Positive Rate	10%	FPR = FP / (FP + TN)
False Discovery Rate	7%	FDR = FP / (FP + TP)
False Negative Rate	4%	FNR = FN / (FN + TP)
Accuracy	94%	ACC = (TP + TN) / (P + N)
F1 Score	95%	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	88%	TP*TN - FP*FN / sqrt((TP+FP)* (TP+FN)*(TN+FP)*(TN+FN))
False Acceptance Rate	4%	FAR=TN/(TP+TN)
False Rejection Rate	2%	FRR=FP/(FP+FN)

7.3 Applicability of Our Research

Smartphone-based complaint system like Report 2 RAB, Police Helpline BD, Bangladesh Bank Complaint etc. can use our approach of detecting lies. Through our application, the authority can determine the important issues with high accuracy from a large number of true and false complaints.Again, social media like Facebook, Twitter, LinkedIn, Google+ etc. can also use our approach for reducing the number of false statuses uploaded by its users. E-recruitment systems are also a part of the applicability of our research. Nowadays a large number of fake applications are being done in our online recruitment applications. Our approach can reduce this tendency from the applicant through identifying lying.

Chapter 8

Conclusion and Future Work

Existing lie detection systems generally demand high cost and hardware overheads. These systems also frequently require an interviewer to ask questions to participants. Thus, the existing systems of lie detection are far from being a ubiquitous solution. Even though smartphones have been widespread in use in many parts of the world exhibiting its ubiquity in nature in recent times, it is yet to be investigated whether the smartphones can be used for the purpose of lie detection or not. As the 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 lie detection mechanism has not been focused in the literature till now.

As a remedy, in this study, we propose a new mechanism for detecting lying through smartphones. To do so, we design a customized survey system having a judiciously chosen set of questionnaire 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 very high accuracy.

Afterwards, based on our findings, we develop a complete 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 42 participants using a new set of questionnaire. We find that the application can provide an average of 96% accuracy.

Subsequently, we develop a dynamic application where a user can set own questionnaire and present that to participants requesting their responses. We present the application to 30 users who conducted survey over 42 participants using their own set of questionnaires. We find 84% accuracy for the dynamic end-user application. Finally, we develop yet another dynamic application using the notion of NMR technique, a widely known fault tolerance technique. The new application enhances the accuracy in lie detection up to 94%, which we confirm by a new set of experiments over eight participants. All these experiments exhibit a gleaming prospect of smartphone for being a near-to-accurate lie detector.

It is worth mentioning that the participants in our study were general 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.

Bibliography

[1] Benussi, "On the effects of lying on changes in respiration," *Archiv fur die Gesamte Psychologie*, 1914.

[2] E. H. Marstonr, "Physiological possibilities in the deception test," *Journal of American Institute of Criminal Law and Criminology*, 1920.

[3] J. A. Larson, "Modification of the marston deception test," *Journal of Criminal Law and Criminology*, 1921.

[4] B. M. D. . W. L. Morris, "Discerning lies from truths: Behavioural cues to deception and the indirect pathway of intuition," *The detection of deception in forensic contexts*, 2004.

[5] P. Ekman and W. Friesen, "Facial action coding system: A technique for the measurement of facial movement," *Consulting Psychologists Press*, 1978.

[6] Statista, "Number of smartphone users worldwide from 2014 to 2020," June 2016. Retrieved on May 5, 2017.

[7] R. W. Picard, "Affective computing," *MIT Press*, 1997.

[8] K. V. Petrides and A. Furnham, "Trait emotional intelligence: Behavioural validation in two studies of emotion recognition and reactivity to mood induction," *European Journal of Personality*, vol. 17, pp. 39 – 57, 2003.

[9] D. Mier, S. Lis, K. Neuthe, C. Sauer, C. Esslinger, and B. Gallhofer, "The involvement of emotion recognition in affective theory of mind," *Psychophysiology*, vol. 47, pp. 1028 – 1039, 2010.

[10] M. O. Sullivan, *Emotional intelligence and deception detection: Why most people can't read others, but a few can*, pp. 215 – 253. Applications of nonverbal communication, Mahwah, NJ: Lawrence Erlbaum Associates Publishers, 2005.

[11] K. Sylwester, M. Lyons, C. Buchanan, D. Nettle, and G. Roberts, "The role of theory of mind in assessing cooperative intentions," *Personality and Individual Differences*, vol. 52, pp. 113 – 117, 2012.

[12] A. Baker, L. Brinke, and S. Porter, "Will get fooled again: Emotionally intelligent people are easily duped by high-stakes deceivers," *Legal and Criminological Psychology*, 2013.

[13] M. Hartwig, P. A. Granhag, and L. Stromwall, "Guilty and innocent suspects strategies during police interrogations," *Psychology, Crime & Law*, vol. 13, pp. 213 – 227, 2007.

[14] T. R. Levine and S. A. McCornack, "Theorizing about deception," *Journal of Language and Social Psychology*, vol. 33, pp. 431 – 440, 2014.

[15] A. Vrij and G. Ganis, "Theoriesin deception and lie detection," *Credibility assessment: Scientific research and applications*, vol. In D. C. Raskin and C. R. Honts and J. C. Kircher, pp. 301 – 374, 2014.

[16] A. Vrij and R. P. Fisher, "Which lie detection tools are ready for use in the criminal justice system?," *Journal of Applied Research in Memory and Cognition*, vol. 5, pp. 302 – 307, 2016.

[17] G. Nahari, A. Vrij, and R. P. Fisher, "Does the truth come out in the writing? scan as a lie detection tool," *Law & Human Behavior*, vol. 36, pp. 68 – 76, 2012.

[18] G. L. J. Lancaster, A. Vri, L. Hope, and B. Waller, "Sorting the liarsfromthe truth tellers:the benefits of asking unanticipated questions," *Applied Cognitive Psychology*, vol. 27, pp. 107 – 114, 2012.

[19] D. C. Raskin and C. R. Honts, *The comparison question test*, pp. 1 – 47. Handbook of polygraph testing, CA: Academic Press, 2002.

[20] R. Nelson, "What does the polygraph measure?," *APA Magazine*, vol. 47, no. 2, pp. 39 – 47, 2014.

[21] R. Nelson, “Redux: What does the polygraph measure? (in 600 words or less),” *APA Magazine*, vol. 47, no. 3, pp. 36 – 37, 2014.

[22] R. Nelson, “Take 3: What does the polygraph measure? (in 250 words or less),” *APA Magazine*, vol. 47, no. 4, p. 60, 2014.

[23] R. Nelson, “Short answer: What does the polygraph measure? (in 150 words or less),” *APA Magazine*, vol. 47, no. 5, p. 27, 2014.

[24] R. Nelson, “Sound-bite: What does the polygraph measure? (in 50 words or less),” *APA Magazine*, vol. 47, no. 6, p. 29, 2014.

[25] E. H. Jang, B.-J. Park, and S.-H. Kim, “Emotion classification based on bio-signals emotion recognition using machine learning algorithms,” *Information Science, Electronics and Electrical Engineering (ISEEE)*, pp. 1373 – 1376.

[26] C. Maaoui, A. Pruski, and F. Abdat, “Emotion recognition for human machine communication,” *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 08)*, pp. 1210 – 1215, Sep 2008.

[27] C.-C. Liu, T.-H. Yang, C.-T. Hsieh, and V.-W. Soo, “Towards text-based emotion detection: A survey and possible improvements,” *International Conference on Information Management and Engineering*, 2009.

[28] Z. Lin, X. Jin, and X. Cheng, “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)*, pp. 79 – 86, 2014.

[29] Z. Lin, S. Tan, and XueQi, “Language-independent sentiment classification using three common words,” *CIKM*, pp. 24 – 28,, October 2011.

[30] S. Liu, W. Zhu, and X. qi Cheng, “Co-training and visualizing sentiment evolvement for tweet events,” *WWW 2013 Companion*, pp. 13 – 17, May 2013.

[31] S. Liu, F. Li, and X. Cheng, “Adaptive co-training svm for sentiment classification on tweets,” *CIKM13*, pp. 2079 – 2088, Oct 2013.

[32] P. Khanna and M.Sasikuma, “Recognising emotions from keyboard stroke pattern,” *International Journal of Computer Applications*, vol. 11, pp. 1 – 5, December 2010.

[33] M. M. Jung, R. Poppe, and M. Poel, “Touching the void -introducing cost: Corpus of social touch,” *ICMI*, pp. 12 – 16, Nov 2014.

[34] Y. Gao, N. B. Berthouse, and H. Meng, “What does touch tell us about emotions in touchscreen-based gameplay?,” *TOCHI*, pp. 39 – 71, Nov 2012.

[35] G. Bauer and P. Lukowicz, “Can smartphones detect stress-related changes in the behaviour of individuals?,” *Work in Progress session at PerCom*, pp. 423 – 426, 2012.

[36] D. Corstange, “Sensitive questions, truthful answers? modeling the list experiment with listit,” *Political Analysis*, vol. 17, pp. 45 – 63, December 2008.

[37] A. Nwaeze, “12 questions ladies often answer with a lie.” Retrieved Febuary 14, 2018 from `https://buzznigeria.com/12-questions-ladies-often-answer-with-a-lie/`.

[38] M. s. Thad Peterson, “100 top job interview questionsbe prepared for the interview.” Retrieved Febuary 14, 2018 from `https://www.monster.com/career-advice/article/100-potential-interview-questions`.

[39] T. Droste, “Don’t get thrown for a loop.” Retrieved Febuary 14, 2018 from `https://www.monster.com/career-advice/article/dont-get-thrown-for-a-loop`.

[40] C. Martin, “Answers to 10 most common job interview questions.” Retrieved Febuary 14, 2018 from `https://www.monster.com/career-advice/article/top-10-interview-questions-prep`.

[41] Jrumple, “Interview question about lying.” Retrieved Febuary 14, 2018 from `https://www.manager-tools.com/forums/interview-question-about-lying`.

[42] R. Rigby, “10 questions you should never answer honestly at work.” Retrieved Febuary 14, 2018 from `https://www.telegraph.co.uk/men/thinking-man/11652018/10-questions-you-should-never-answer-honestly-at-work.html`.

[43] IBM, "Intelligent information systems." Retrieved Febuary 14, 2018 from `http://www.almaden.ibm.com/software/quest/resources/`.

[44] Y. Theodoridis and M. Nascimento, "Generating spatiotemporal datasets," *ACM SIGMOD Record*, 2000.

[45] S. Ross, "A first course in probability," *Prentice Hall*, 1997.

[46] Y. Pei and O. Zaiane, "A synthetic data generator for clustering and outlier analysis," *ERA*, 1998.

[47] D. Hogan, B. V. Maruthachalam, R. C. Geyer, and A. Kusalik, "Weseqminer: A weka package for building machine-learning models for sequence data," *bioRxiv*, p. 217802, 2017.

[48] R. Seidlová, J. Poživil, J. Seidl, and L. Malecl, "Synthetic data generator for testing of classification rule algorithms," *Neural Network World*, vol. 27, no. 2, p. 215, 2017.

[49] J. Chandrasekaran, H. Feng, Y. Lei, D. R. Kuhn, and R. Kacker, "Applying combinatorial testing to data mining algorithms," in *Software Testing, Verification and Validation Workshops (ICSTW), 2017 IEEE International Conference on*, pp. 253–261, IEEE, 2017.

[50] D.-C. Li, Q.-S. Shi, and M.-D. Li, "Using an attribute conversion approach for sample generation to learn small data with highly uncertain features," *International Journal of Production Research*, pp. 1–14, 2018.

[51] J. A. Hernández-Castaño, O. Camacho-Nieto, Y. Villuendas-Rey, and C. Yáñez Márquez, "Experimental platform for intelligent computing (epic)," *Computación y Sistemas*, vol. 22, no. 1, 2018.

[52] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms," in *Proc. of KDD-2013*, pp. 847–855, 2013.

[53] "Weka 3: Data mining software in java," 2016. Last accessed on 01 March, 2017 `http://www.cs.waikato.ac.nz/ml/weka/index.html`.