A Behavioral Biometrics-based Trust Framework for Continuous Crowdsensing Recruitment

Ruba Ayman Nasser

MSc. Thesis

December 2021

A thesis submitted to Khalifa University of Science and Technology in accordance with the requirements of the degree of MSc. in Electrical and Computer Engineering in the Department of Electrical Engineering and Computer Science.

A Behavioral Biometrics-based Trust Framework for Continuous Crowdsensing Recruitment

by

Ruba Ayman Nasser

A thesis submitted in partial fulfillment of the requirements for the degree of

MSc. in Electrical and Computer Engineering

at

Khalifa University

Thesis Committee

Dr. Rabeb Mizouni (Main Advisor),
Khalifa University

Dr. Shakti Singh (Co-Advisor),
Khalifa University

Dr. Hadi Otrok (Co-Advisor),
Khalifa University

Prof. Hassan Barada (RSC Member 1),
Khalifa University

Dr. Kamal Taha (RSC Member 2),
Khalifa University

December 2021

Abstract

Ruba Ayman Nasser, "**A Behavioral Biometrics-based Trust Framework for Continuous Crowdsensing Recruitment**", MSc. Thesis, MSc. in Electrical and Computer Engineering, Department of Electrical Engineering and Computer Science, Khalifa University of Science and Technology, United Arab Emirates, December 2021.

The inevitable trend of the large spread of smartphones and smart devices around the world has opened the door for a new sensing paradigm known as Mobile Crowdsensing (MCS). By recruiting workers with mobile devices to perform tasks over a cloud based MCS platform, organizations can collect sensing data in less time and at lower cost. One of the challenges faced in MCS systems which could prevent the task requester from obtaining reliable information is the presence of malicious workers who join the sensing task by impersonating other workers' identities. Identifying whether the data reports were submitted by a genuine user or an impersonator can help MCS systems exclude distrusted workers from the sensing task and therefore improve the quality and integrity of the submitted sensing reports. However, since MCS systems collect data from users while ensuring that their privacy is protected, the submitted reports cannot be linked to the workers which makes detecting impersonators in the system challenging. Behavioral biometrics refers to the unique behavioral traits that can be used to authenticate users based on how they naturally perform a specific activity. This work proposes a touch screen input behavioral biometrics-based trust framework that can support a reliable recruitment process in continuous MCS tasks. Using the touch screen input data collected from users in previous sensing tasks, unique machine learning models are built for each MCS worker which are then used to detect impersonators in the group during future sensing tasks. The proposed approach integrates the trained machine learning models with a dynamic continuous recruitment system taking into consideration the uncertainties accompanied with using the machine learning models. Simulations show that the proposed system improves the quality of the submitted reports when compared to a benchmark that only relies on users' historical performance.

Indexing Terms: quality of information, behavioral biometircs, touchscreen dynamics.

Acknowledgement

I would like to express my gratitude to everyone who helped me and supported me in writing this thesis, especially my advisors: Dr. Rabeb Mizouni, Dr. Shakti Singh and Dr. Hadi Otrok, who provided me with the guidance and support I needed throughout these two years.

I would like also to thank my examiners: Prof. Hasan Barada and Dr. Kamal Taha, for their valuable feedback throughout my progress.

Special thanks to Eng. Menatalla Abouaouf and Dr. Maha Kadadha for supporting me and taking their time to help me especially during stressful times.

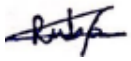
Last but not least, I wish to acknowledge the support and great love of my parents, my husband and my family who always believed in me and kept me going on.

Declaration and Copyright

Declaration

I declare that the work in this thesis was carried out in accordance with the regulations of Khalifa University of Science and Technology. The work is entirely my own except where indicated by special reference in the text. Any views expressed in the thesis are those of the author and in no way represent those of Khalifa University of Science and Technology. No part of the thesis has been presented to any other university for any degree.

Author Name: Ruba Ayman Nasser

Author Signature: 

Date: 11/12/2021

Copyright ©

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Nowadays, the large spread of smartphones and smart devices around the world has led to the emergence of Mobile Crowdsensing (MCS). MCS is a new sensing paradigm where user-paired devices are recruited by the management platform to perform a sensing task in a specific area. A typical MCS system consists of a task requester, the management platform and the MCS workers. Its deployment complements the existing Internet of Things (IoT) sensing solutions in supporting the vision of smart cities and improving the quality of citizens' life. In fact, it is viewed as an important solution for building smart cities since human mobility and intelligence offer higher coverage and deeper contextual understanding of the sensing tasks. One of the important factors leading to the evolution of MCS is the cloud computing environment, where data is stored and processed over the Internet. Task requesters can benefit from the ease of access to shared resources and the efficient data management offered by the cloud, in addition to human mobility, to ensure high quality sensing at low cost and in an efficient and timely manner. A typical MCS paradigm covers various aspects including sensing the required data, communicating the sensed data between the workers and the cloud , user recruitment and task allocation strategies, which aim to maximize the quality of infor-

mation (QoI) and allocate appropriate resources for the sensing tasks[1]. MCS tasks can be classified into one-time sensing tasks and continuous sensing tasks. One-time sensing tasks are usually employed in event-based MCS systems where the sensing data is submitted once a certain event takes place, like detecting and reporting a car accident, monitoring bus arrival time, comparing prices of goods and health care applications. On the other hand, continuous sensing tasks require collecting information continuously in the area of interest (AoI) for the total sensing period, to accurately study the phenomena that task publisher is interested in. For example, monitoring noise, air pollution, and characterizing the coverage of WiFi intensity, all required data to be collected continuously for a specific period of time. Once the tasks are publicized, user selection is usually performed based on their location, their willingness to perform the task and/or their reputation [1]. MCS tasks often seek the cooperation of multiple workers to collect data that cannot be collected only by a single worker. For example, to monitor air quality in different cities, a single user can only collect some of the data needed by the application since the locations that the user can reach can be limited [2]. In order to select the most appropriate group to perform the sensing task, the recruitment system needs to consider the participants' collective QoI which can be affected by parameters related to the AoI,the device and the user. In continuous sensing tasks, these parameters may change during the period of sensing due to group members leaving the AoI or losing connectivity with the management platform before the end of the sensing task. Therefore, in order to ensure the group is meeting the required QoI by the task publisher during the sensing task, the recruitment system needs to add and remove members to the group continuously until the required QoI is met [3].

1.1 Motivations and Problem Description

For MCS systems, trusting users is crucial for the well-functioning of the system [4]. Incorporating workers' reliability into the data aggregation procedure can make the

system more robust against data poisoning attacks which are conducted by malicious workers in the crowdsensing system [5]. Several mechanisms for evaluating users' trustworthiness to ensure high data quality and integrity in MCS were proposed. Overall, trust can be established by learning from previous experiences, observations and opinions from other entities. The system usually checks and rates users' performance in previous tasks, selects a subset of them to perform a new task and then after the task is completed, the users' old ratings get updated [6]. However, an adversary may still succeed at tricking the management platform into getting recruited to the sensing task by impersonating another worker's identity. Therefore, a more reliable recruitment mechanism should also consider monitoring participant's behavior during the task in order to remove distrusted participants from the group and replace them with more trustworthy ones. A dynamic recruitment system for continuous sensing tasks was simulated in an environment where all participants are genuine versus an environment where some of them are impersonated, to illustrate the impact of the presence of impersonators in the group. The simulation was performed for a task with a required QoI of 2.7 using the stability-based group-based recruitment system proposed in [3]. Based on the value of the the participants' AoI-related parameters, user-related parameters and device related parameters, the expected QoI evaluated by taking the weighted sum of all these parameters must be at least 2.7. The stability-based group-based recruitment system (stability-based GRS) is a recruitment system proposed for continuous sensing tasks which considers participants mobility during the selection process to ensure high coverage and the task requester's desired QoI. The system employs genetic algorithm to perform the initial recruitment and select the most stable group of participants taking into consideration participants' mobility patterns. Once the initial group is recruited, the proposed system continuously adds and removes members to the group to ensure that the publisher's required QoI is met [3]. In this example, the dataset of mobility traces in the city of Cologne, Germany, was used for both truthful and untruthful environments. The simulation was performed for a sensing task with AoI boundaries (10000 to 15000)x(10000 to 15000) for a duration of 80 seconds. The sensing period was di-

vided into 4 equal sensing intervals of length 20 seconds each. Snapshots of the selected groups in the second and fourth intervals in both environments is shown in figures 1.1 and 1.2. For the untruthful environment, 20 % of the population was randomly chosen to be the impersonators. Figure 1.1 shows the users selected 40 and 80 seconds after the task has started in a truthful environment. At t=40 seconds, 19 users were needed in order to achieve a QoI of 2.7, whereas at t=80 seconds, two additional workers were needed (W262 and W288) to keep the QoI of the group from falling below 2.7. On the other hand, in an untruthful environment, the group members at t=40 seconds included two misbehaving workers who are impersonating the identities of W580 and W435 as shown in figure 1.2. Additionally, at t=80 seconds, an additional misbehaving worker who is impersonating the identity of W351 was added to the group. We use the negation symbol before the identity of the worker to denote that this is not the true identity of the worker, i.e, the worker is not genuine. In the stability-based GRS, the QoI of the group is evaluated based on AoI-related parametetrs, Device-related parameters and user-related parameters. These parameters are further explained in section 4.1.1 . It can be seen from figures 1.1 and 1.2, that although other truthful workers were available in the AoI, the presence of the misbehaving workers prevented these truthful workers from getting selected. The QoI of the group at each time point in a truthful environment

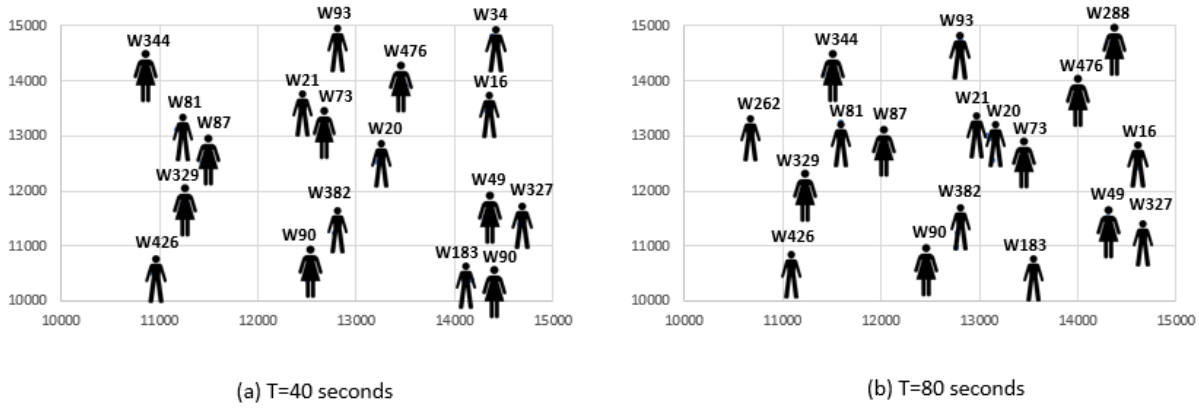


Figure 1.1: Recruited groups in a truthful environment at different time points

is compared against the obtained values in an untruthful environment in Table 1.1. The expected QoI is evaluated based on the selected workers without the detection of the impersonators, while the achieved QoI is computed when considering only genuine users.

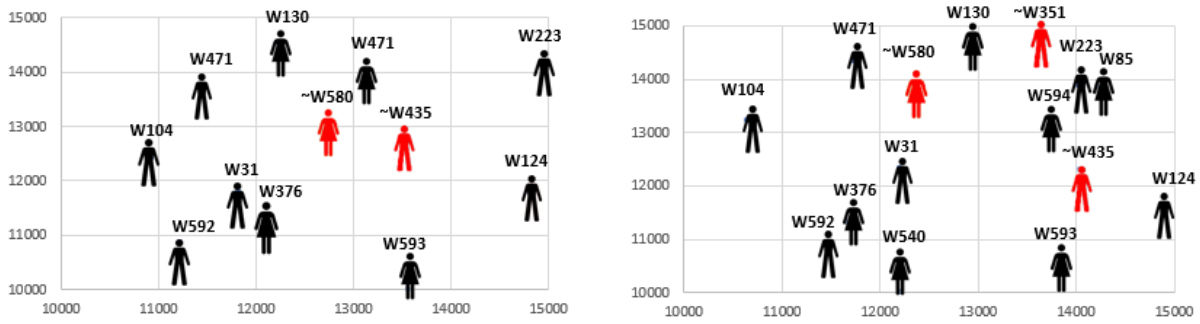


Figure 1.2: Recruited groups in an untruthful environment at different time points

As can be seen from the table, the achieved QoI falls below the required value of 2.7 in both sensing intervals. In addition, the achieved QoI is less than than the expected QoI by 8.5% and 17% for both intervals respectively in the untruthful environment. In the

Table 1.1: Expected vs. Achieved QoI at each intervals in truthful and untruthful environments.

Time	Truthful Environment	Untruthful Environment	
	Achieved QoI	Expected QoI	Achieved QoI
40 seconds	2.724879	2.729821	2.516169
80 seconds	2.725578	2.739971	2.34095

untruthful environment, a total of three impersonators contributed sensing data throughout the period of sensing. However, these impersonators cannot be trusted since they could have malicious intentions and wish to degrade the performance of the system by submitting poisoned data. At T=40 seconds and T=80 seconds, 13% and 20% of the sensing reports in each interval respectively are considered unreliable since they were submitted by non genuine users . In addition, since the system can not detect the impersonators in the group, untruthful workers are still getting paid. As shown in Table 1.2, 26 % of the total cost is given to the misbehaving workers in the group in the untruthful environment. The payment evaluation is further explained in section 4.1.2.

The main goal of this study is to propose an approach which can be adopted in continuous sensing tasks recruitment systems to maintain the required QoI value by addressing the following research question:

How can continuous sensing tasks' recruitment systems detect and eliminate impersonators during the sensing period?

Table 1.2: Total payments in truthful and untruthful environments

Truthful Environment	Untruthful Environment	
Payment	Truthful Payment	Untruthful Payment
357.62	246.5425	86.98

1.2 Methodology and Contributions

To address the problem of recruiting impersonators in continuous sensing tasks , a novel approach for verifying that the submitted data reports are from genuine workers is needed to ensure high data quality and integrity during the sensing task. Since MCS relies heavily on data collection, different machine learning techniques can be used in order to improve the overall performance and trustworthiness of the system. Using the data collected from users during sensing tasks, the system can build unique behavioral models for each user and use them in future tasks in order to predict or detect misbehavior [1]. Today's smart devices are equipped with different sensors that can be used to collect data from users which can serve as identifiers while the user is doing the sensing task [7]. The unique behavioral traits that describe how a user performs a specific activity such as walking, typing or interacting with the smartphone's touch screen is referred to as behavioral biometrics [8]. This work proposes a behavioral biometrics-based classification framework for continuous mobile crowdsensing recruitment that can detect and eliminate impersonators from the recruited group. The main contribution of this work can be summarized as follows:

- Leverage machine learning to build unique behavioral biometrics models for each MCS worker that can be used to detect impersonators in the group during the sensing tasks.

- Propose a behavioral biometrics-based trust evaluation mechanism which can be adopted in MCS recruitment solutions to remove impersonators from the group during continuous sensing tasks, taking into account the uncertainties accompanied with the predictions made by the machine learning model.

- Use the trained models to improve the efficiency of the worker selection by min-

imizing risks and potential attacks on the systems as only trusted workers get involved in the task fulfillment.

MCS systems usually consist of three main entities: task publishers, the management platform and the mobile workers as shown in Fig. 1.3 . After the management platform receives the sensing task from the task requester, it starts recruiting the appropriate participants that can perform the sensing task. Once the task is completed, the management platform evaluates and aggregates the submitted reports and forwards the aggregated sensing reports back to the task requester [9]. The main goal of this work is to protect continuous MCS systems from impersonation attacks by building unique behavioral models that can identify genuine and non-genuine users from the way they interact with their smartphones' touchscreens. A high level architecture of the proposed system is illustrated in Fig. 1.3.

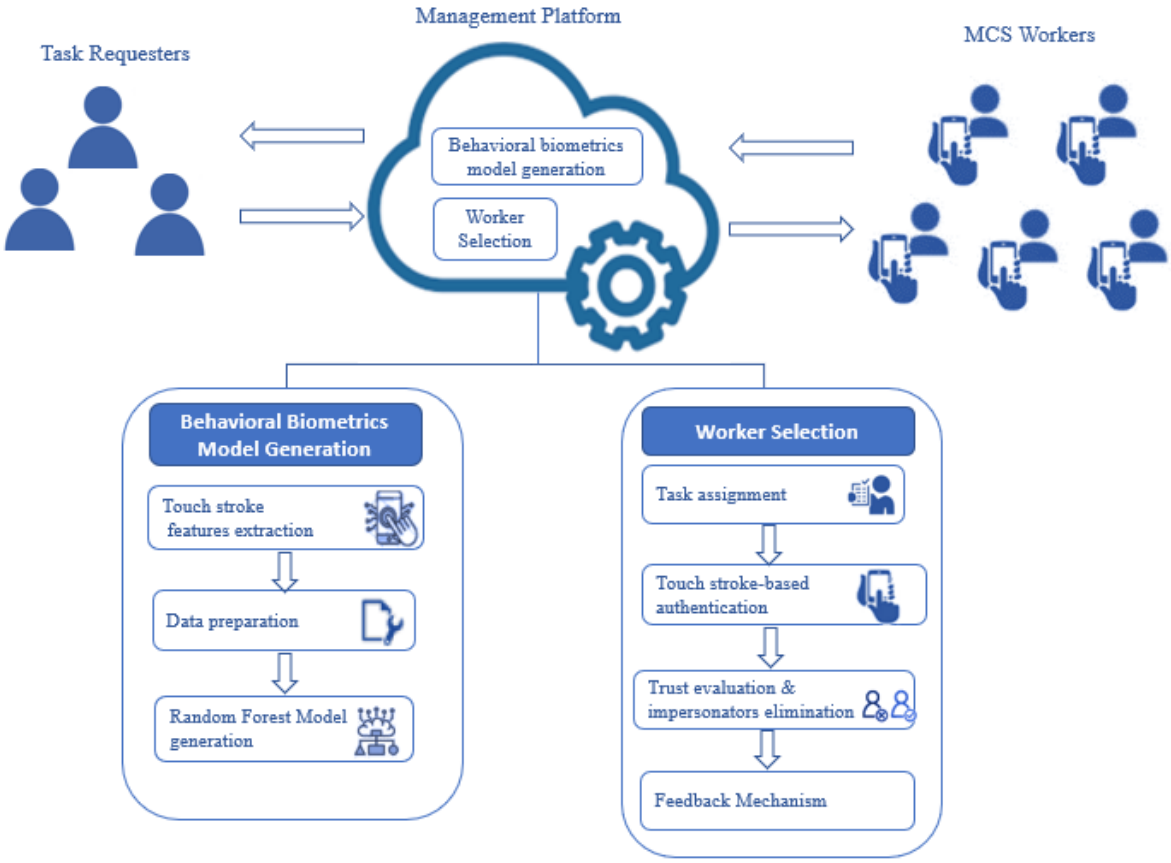


Figure 1.3: High level architecture of the proposed system

In order to incorporate worker's behavior predication in the recruitment process,

the crowdsourcing platform is redesigned to include two main modules: Behavioral Biometrics Model Generation, and Worker Selection [9] which are summarized below:

- **Behavioral Biometrics Model Generation**: The extracted features obtained from the users' touchscreen input data are used to generate a customized training model for unique profiling of a worker. The classification problem is treated as a binary classification problem, therefore prior to training, touch features from all other users are randomly selected to model the impersonator's behavior. Then, the data is labeled based on whether the touch stroke features belong to the user or not.

- **Worker Selection**: In this module, the previously generated behavioral biometrics models are used to classify each touch stroke feature either to belong to a genuine user or to an imposter. After that, a decision is made to keep or remove the participant from the sensing task based on the predictions made by the trained model taking the uncertainties in the predictions made by the model into consideration. Finally, a feedback mechanism is adopted in response to whether the worker was detected as a genuine worker or an impersonator.

1.3 Thesis organization

This thesis is organized as follows:

- **Chapter 1:** In this chapter, a general overview of MCS is provided followed by a description of the main challenge tackled in this thesis and an illustrative scenario of the problem. At the end of the chapter, the main contributions of this work are stated and the adopted methodology to solve the problem is described in general.

- **Chapter 2:** In this chapter, a background about MCS architecture and behavioral biometrics based authentication is first provided. After that, a literature review is provided for the work done on ensuring high data quality in MCS systems, in addition to the work done on authenticating mobile phone users using behavioral biometrics.

- **Chapter 3:** This chapter describes the data used in training the behavioral biometrics machine learning models in this work as well as the approach adopted for generating these models. At the end of the chapter, the performance of the supervised machine learning models are evaluated using precision, recall and f1score metrics.

- **Chapter 4:** This chapter describes the overall proposed approach for integrating the trained behavioral biometrics models in continuous MCS recruitment systems. The different attributes used to represent a worker are first explained, then the recruitment mechanism adopted is discussed ,taking into consideration the uncertainties accompanied with the predictions made by the model.

- **Chapter 5:** In this chapter, the trained machine learning models in chapter 3 are used in continuous sensing tasks for detecting and eliminating impersonators every sensing interval. Simulation results of the proposed approach are provided and compared with a benchmark.

- **Chapter 6:** This chapter includes a summary of the final outcomes of the thesis and lists the challenges that could be faced by the proposed system.

CHAPTER 2

Background and Related Work

This chapter includes two main sections: Background and literature review. In the background section, a general overview of the MCS architecture is explained first,then, behavioral biometrics based authentication is briefly discussed. After that, a literature review is provided for the work done on ensuring high data quality in MCS systems, in addition to the work done on authenticating mobile phone users using behavioral biometrics.

2.1 Background

2.1.1 MCS Architecture

In MCS systems, the data sensed using participants' devices travel through multiple layers before it gets aggregated and processed to estimate the ground truth. As shown in Fig. 2.1, an MCS system can viewed as a 4-layered architecture. Starting from the bottom, the sensing layer is responsible for sensing the required data by the users either using sensors embedded in participants' devices such as motion sensors and GPS sensors or using external sensors designed for specific purposes which can be connected

to users' smartphone devices such as gluten sensor that can detect gluten in food. In terms of user involvement, the data collection process can either take place in an opportunistic or in a participatory manner. In opportunistic sensing, users are not asked to perform a specific action, however, an application is run in the background and the data collection is performed automatically, such as monitoring users' mobility patterns using the GPS sensors embedded in their smartphones. On the other hand, in participatory sensing, a user is required to perform a task at a specific time and location and then users are rewarded based upon the quality of the data they submit. The second layer from the bottom is the communication layer which is responsible for transmitting the sensed data from the application layer to the cloud. After that, the third layer ,which is the data layer, includes storing and processing the sensed data in the cloud.

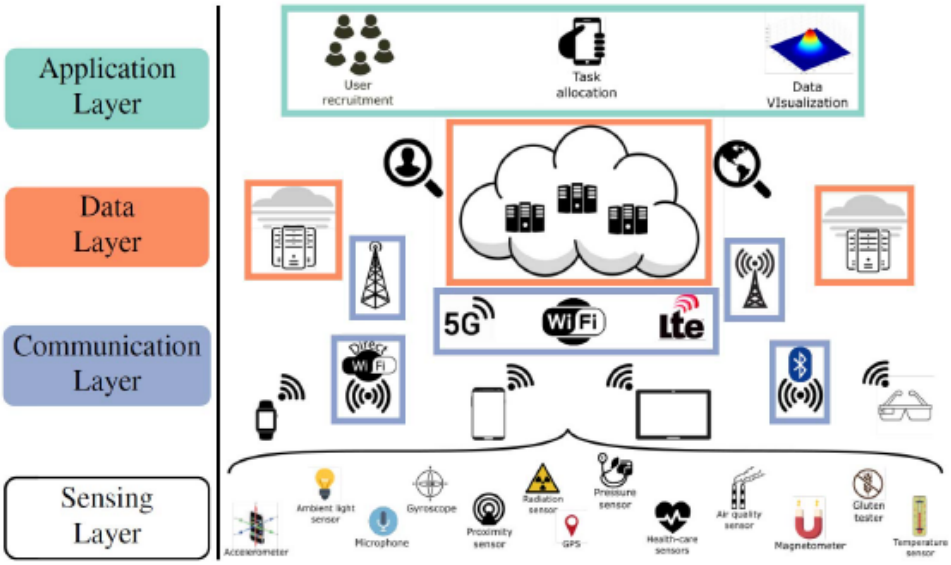


Figure 2.1: MCS architecture

Finally, in the application layer, user recruitment for each sensing task takes place such that the highest quality of information is achieved. Different incentives can be used to motivate users to participate in the sensing task and could be either in the form of a monetary reward, a service or for entertainment. The main goal of the user recruitment strategies is to minimize the overall cost while maintaining the desired level of quality of information. In addition, the application layer also includes task allocation strategies implemented in multi-task scenarios where users are allocated to multiple tasks pub-

lished by different task requesters. The main goal of these strategies is to allocate the tasks for users taking into consideration the different locations and timing requirements of these tasks [1].

2.1.2 Authentication using Behavioral Biometrics

Thanks to authentication, any entity can ensure that a certain user is the one she/he claims to be. Authentication is usually achieved through the use of identity credentials, which could be either something the user knows, like passwords, something the user has, like identification cards, or something representing the user's unique biometric characteristics like fingerprint traces. However, in such authentication approaches, the chances of impersonation are high since users get authenticated one time only at the beginning of the session[7]. If the authentication information is stolen or compromised, imposters will be able to illegally participate in the sensing task from another user‘s account which could lower the safety of the system. For example, in a ride-sharing platform , if registered drivers get impersonated, rides can be assigned to unauthorized individuals which could be dangerous for the riders. Additionally, existing solutions require explicit input or actions, which makes them obtrusive for users since they require users' attention and could cause a distraction from the sensing task [9]. Unlike traditional authentication, where authentication decision is taken at a given moment, continuous authentication (CA) monitors user actions every point in time during a session and determines whether or not the user is legitimate [7]. The unique behavioral traits that can be used to continuously profile users based upon their natural interactions and without having to constantly interrupt them during the session is referred to as behavioral biometrics. The most commonly used behavioral biometrics in literature include keystroke dynamics, walking gaits and touchscreen dynamics where users are authenticated by the way they type, walk or interact with the smartphone's touch screen respectively [9].

2.2 Ensuring Data Quality and Integrity in MCS systems

Ensuring high quality sensing data and high integrity is an important problem that should be addressed in MCS systems. To tackle this challenge, the presence of malicious or selfish users providing low quality data should be taken into account at the recruitment stage. Multiple mechanisms have been proposed in literature to establish trust in MCS settings [6]. In [10], a framework to assign tasks to the best group of users who will return high quality reports within the required sensing period was proposed. The quality of information evaluated for each worker depends on the reputation, the confidence that a worker will complete the task within a given period of time and the distance between the worker and the given task. The reputation parameter computed by the MCS system depends on the historical performance of the worker. It considers that a task was performed successfully if their answers are equal to the estimated ground truth value of the system [10]. In [11], a reputation system was proposed to assess and predict the trustworthiness of users via endorsement links. Every endorsement link is associated with a weight which reflects one user's confidence in another user. The user's reputation is evaluated based on the requester's evaluation for the task that the user has performed. The trustworthiness of worker contribution is predicted based on : the workers' reputation score, which reflects their reliability and historical performance, and the endorsement impact from the worker's endorsers, which consists of: the reputation of the endorsers, the degree of endorsement and the heterogeneity of tasks [11]. In [12], a dynamic-trust-based recruitment framework was proposed which calculates the overall trust based on real-time direct trust and indirect trust. Real-time direct trust refers to the trust of the task requester in the worker in the recent past. The latest interaction record is given more weight than previous records in this trust evaluation method. On the other hand, the indirect trust evaluation method relies on collecting feedbacks from the task requester after the task is over for other task requestors as a reference in future interactions [12]. In [13], the authors proposed a framework to evaluate trust value of

the users' sensing reports in participatory sensing networks based on various factors such as time of sensing, location of sensing, sensor mode and user's traveling mode. Using all the collected reports for a task,a similarity factor is assigned to each report and once then the trust value is calculated. After that, the reputation feedback level for the participant is generated and encrypted within a reputation feedback coupon [13].

2.3 Keystroke Dynamics Behavioral biometrics

In keystroke dynamics, features describing the typing rhythm such as the keystroke length, the pressure exerted on each key while typing and the time difference between consecutive strokes are used [9]. The performance of a range of anomaly detection algorithms employed to authenticate users based on keystroke dynamics was evaluated and compared in [14]. The top-performing detectors found were Manhattan, Nearest Neighbor and Outlier Count (z-score) [14]. Additionally, two binary classifiers: BayesNet and Random forest were used by [15] to perform authentication using keystroke features along with features describing the user's phone holding behavior obtained from the smartphone's built in sensors. The proposed scheme showed acceptable authentication rates with data that was collected in six different user positions: sitting, standing, walking, walking upstairs, walking downstairs and lying on the sofa [15]. Although continuous authentication based on keystroke dynamics provides unobtrusive data collection, the variability of typing behavior is expected to appear across different sittings which makes this type of behavioral biometrics scenario dependent. In addition, the flexibility of the input text requires the need to gather as much typing input as possible, which translates to longer waiting time before the authentication can be performed efficiently [16].

2.3.1 Walking Gait Behavioral Biometrics

Multiple works in literature have shown that individuals can be recognized by their gait provided that proper motion measurements are taken. Walking gaits behavioral bio-

metrics refers to the characteristic and mannerism in which an individual walks [17]. Today's smartphones and smart devices are equipped with built in motion sensors such as accelerometers and gyroscopes which contribute data that can be used to extract unique features using the prevalent signal processing techniques. In [18], a convolutional neural network-based deep learning model was proposed to identify smartphone users in crowd sensing systems based on the data produced by the accelerometer sensors in their smartphones. It was concluded that the Fourrier transform is a simple but very powerful technique in the feature extraction process which has improved the accuracy of the model[18]. In [19], a deep neural network based scheme which relies on the unique physical features of WiFi signals during the daily activities of mobile users was proposed. The system extracts 6 time domain features and 3 frequency domain features from both the amplitude and the phase channel of the channel response of WiFi signals. The extracted features are then used in a three-layer stacked autoencoder to perform activity recognition and then user authentication [19]. In order to accelerate the authentication process without having to perform feature extraction at an earlier stage,[20] introduced a deep learning approach that self-learns the necessary network traffic features to authenticate the MCS users. Using a stacked autoencoder, the first layer learns first-order features which are then used in the second layer which learns the features corresponding to the patterns from the previous features [20].

Continuous authentication based on walking gaits can be affected by several internal factors including psychological conditions and illness as well as external factors such carrying a load and the type of footwear. In addition, prior studies and experiments were conducted in a controlled environment, which is not the case in the real physical world. As a result, it is expected that models trained using a dataset in a certain situation would introduce bias when applied to other situations [9].

2.3.2 Touchscreen Dynamics Behavioral Biometrics

In touchscreen dynamics behavioral biometrics, features related to the on-screen sliding movements that represent the user's unique interaction patterns with the smartphone's

touchscreen are used.

A real-time reauthentication scheme for smartphones using touchscreen dynamics behavioral biometrics was proposed in [21]. Five machine learning algorithms were employed to authenticate users including decision tree, naive Bayesian, K-nearest neighbor, logistic regression and SVM. SVM classifier was found to be the best suited for authentication with lower equal error rate (EER) and better performance than the other machine learning methods. The data used for training and testing was obtained from users as they used their smartphones in a routine manner over a period of one month [21]. The performance of ten touch-based authentication classification algorithms were evaluated in [22]. The best performing algorithm found with the lowest EER was the logistic regression machine learning algorithm. The data used was obtained from users as they answered multiple choice questions on their smartphones over two sessions that were at least one day apart [22]. In [23], SVM was adopted to perform touch stroke based authentication using data collected from users while normally using their smartphones during a 15 minute session for 21 days. To model a valid user in the authentication classifier, the user's data during the previous 20 days were used. On the other hand, in order to model the attacker in the authentication classifier, data obtained from users who were selected randomly from the remaining users in the dataset was used. Overall, an improvement in the average error rate was observed as the number of the randomly selected users to model the attacker increased [23]. In [24], touch stroke features were used in two different classifiers, K-neareset-neighbors (KNN) and support vector machine (SVM) to authenticate users in three different experimental settings: inter-session authentication, inter-week authentication and intra-session authentication.

The data used was collected from users as they performed two different tasks: a reading task and an image comparison task. Overall, the authentication difficulty seemed to increase with the increase in the time difference between training and testing, and SVM always achieved a lower EER than KNN algorithm [24]. Authentication based on touch operations provides a natural way to collect user interaction data. Each user generates unique touch patterns, which depend on the application being used.

Table 2.1: summary of related work

Reference		Behavioral biometrics-based authentication			Trust based recruitment mechanism
		Keystroke dynamics behavioral biometrics	Walking gait behavioral biometrics	Touch screen dynamics behavioral biometrics	
[3]		×	×	×	✓
[10]		×	×	×	✓
[11]		×	×	×	✓
[12]		×	×	×	✓
[13]		×	×	×	✓
[14]		×	×	×	✓
[15]		✓	×	×	×
[16]		✓	×	×	×
[19]		×	✓	×	×
[20]		×	✓	×	×
[21]		×	✓	×	×
[22]		×	×	✓	×
[23]		×	×	✓	×
[24]		×	×	✓	×
Proposed system		×	×	✓	✓

Therefore, this type of authentication is well suited to protect against access of unauthorized individuals to important mobile applications [9].

Multiple solutions have been proposed to establish trust in MCS by considering the historical performance of workers in previous tasks. However, In continuous sensing tasks,users are required to collect many readings to over a given time interval which makes recruitment challenging [3]. Therefore, a more robust recruitment mechansim for continuous sensing tasks needs to consider monitoring the behvior of users during the sensing period, in order to increase the trust of the system in the recruited participants. This work leverages touchscreen dynamics behavioral biometrics to monitor the behavior of the participants recruited for continuous sensing tasks. Based on the real-time prediction made by the touchscreen dynamics authentication models, the proposed system removes users from the recruited group in case the trust of the system in these users is too low. A summary of the related work is shown in Table 2.1.

CHAPTER 3

Touchscreen Input Behavioral Biometrics

This chapter describes the data used in training the behavioral biometrics machine learning models as well as the approach adopted for generating these models. At the end of the chapter, the performance of the supervised machine learning models are evaluated using precision, recall and f1 score metrics.

3.1 Data preparation

This work uses supervised machine learning to predict whether the generated touch strokes during the sensing task belong to a genuine user or to an impersonator. Hence, any dataset used containing n records should be organized as $(\vec{x}, y) = (x_1, x_2, ..., x_m, y)$, where m is the number of features , $\vec{x}$ is the features' vector and $y \in Y$ is the dependent variable which represents the label given to $\vec{x}$ [9]. In this work the label is binary where y=1 indicates that the stroke was generated by a genuine user and y=0 indicates that the stroke was generated by an impersonator [25]. The overall workflow is summarized in Fig. 3.1. As shown in the figure, for each worker, data for both the genuine user class and the impersonator class in the training dataset are obtained from the same

worker's session 1 stroke features and other workers' session 1 stroke features selected at random. A sliding window technique is adopted in order to prepare the data for the training phase and finally the worker's machine learning model is trained using Random Forest.

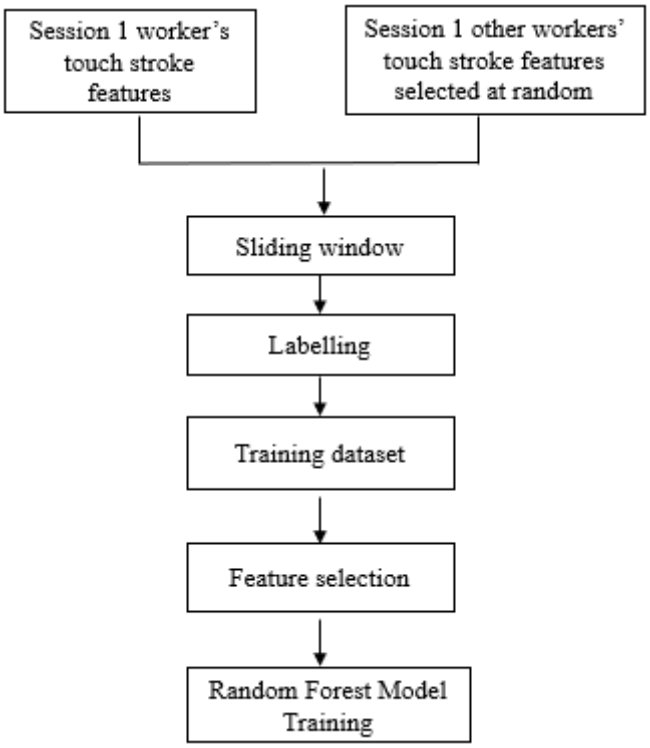


Figure 3.1: Flowchart depicting the experimental workflow

3.1.1 Features Description

A single touch stroke can be defined as the sequence of touch data that begins with touching the screen and ends with lifting the finger. One stroke s is a trajectory encoded as a sequence of vectors $s_n = \{x_n, y_n, t_n, p_n, A_n, o_n^f, o_n^{ph}\}$. The parameters used to model a stroke vector are described in table 3.1 . Using basic navigation maneuvers for each touch stroke, different features can be extracted which could be either temporal features, spatial features, geometric features or statistical features. This work uses the Touchalytics dataset which includes 30 touch stroke features collected from 40 different users who were asked to spot differences between pairs of similar images over two sessions. Table 3.2 describes all the features used in this work. The features demon-

Table 3.1: Parameters used to describe a single touch stroke

Symbol	Definition
x_n	x coordinate on the screen (number of pixels along the horizontal axis)
y_n	y coordinate on the screen (number of pixels along the vertical axis)
t_n	absolute time stamp of the recorded action (in ms)
p_n	pressure on the screen
A_n	screen area covered by the finger
$o^f{}_n$	orientation of the finger with respect to the screen
$o^{ph}{}_n$	Orientation of the phone (1 for landscape mode and 2 for portrait mode)

strate different aspects of users' behavior as they scroll up and down or right and left. For example, the median velocity of the last five points of the trajectory can distinguish between users who stop the screen before lifting their finger, and users who lift their finger while it still has a nonzero velocity, leaving the screen moving even after lifting the finger. Moreover, the largest deviation can give an indication to whether the user is left or right handed depending on whether the deviation is to the left side or to the right side [25].

3.1.2 Data pre-processing

In touchscreen based behavioral biometrics, touch stroke features obtained from one session can be used to build unique models for each worker. Therefore, in the training phase, every classifier is trained using the data obtained from the first session whereas in the testing phase, the data obtained from the second session is used. A snapshot of the training data for user 1 is shown in Table 3.3. The table shows five values of six different stroke features that belong to the user with ID 1.

After splitting the training and testing datasets, the data for every worker was filtered based on their unique IDs. Multiple consecutive strokes were then combined together using a sliding window in order to obtain a better classification accuracy. In the sliding window method, a window of length n moves over the data sample by sample, and computes the average of the data in the window. To model the impersonator's touching behavior, touch strokes are randomly selected from other users in the dataset such that the number of strokes of the genuine user equals the number of strokes of the impersonator [25]. Finally, the combined stroke features for both classes in the training dataset

Table 3.2: Touch stroke features used in this work.

	Features
Temporal Features	Time difference between two consecutive strokes
	Stroke Duration
Geometric Features	Screen area covered by the finger in the middle of the stroke.
	Length of the trajectory: sum of pairwise distances through the stroke
	Mean length for each $(x_n, y_n), (x_{n+1}, y_{n+1})$ pair
Spatial Features	x start position of the stroke
	y start position of the stroke
	x end position of the stroke
	y end position of the stroke
	End to end distance of the stroke
	Direction of the end-to-end distance of the stroke
	Mean direction for each $(x_n, y_n), (x_{n+1}, y_{n+1})$ pair
	Average velocity of the stroke
	Encoded direction of moving the screen: 1 up,2 down, 3 left, 4 right
	Pressure exerted by the finger in the middle of the stroke
	Ratio between end-to-end distance and length of the trajectory
	Change of the finger orientation during the stroke
	Phone orientation
Statistical features	20^{th} percentile of the pairwise velocity
	50^{th} percentile of the pairwise velocity
	80^{th} percentile of the pairwise velocity
	20^{th} percentile of the pairwise acceleration
	50^{th} percentile of the pairwise acceleration
	80^{th} percentile of the pairwise acceleration
	Median of the last three points of the velocity
	Largest deviation from end-to-end line: the maximum distance between the direct end to end line and the line of the trajectory
	20^{th} percentile of the deviation from end-to-end line
	50^{th} percentile of the deviation from end-to-end line
	80^{th} percentile of the deviation from end-to-end line
	Median acceleration over first 5 points

are labelled.

3.1.3 Features Selection

Not all touch stroke features can be used to uniquely distinguish a worker from other users across multiple sessions in the same way. In reality, each worker may have some specific behavioral features which are more important than other features. Considering the unimportant features in the training phase unnecessarily increases the computational complexity of the problem and can cause over fitting. To avoid that, permutation feature importance (PFI) is used to compute the importance score of the features for each worker. First, each model is trained using the original training data. Then, the models are trained another time using the same data but after shuffling the records of the feature

Table 3.3: Snapshot of the training data of worker 1

ID	Inter stroke time	Stroke duration	Length of the trajectory	Average velocity	Median acceleration at first 5 points	Mid-stroke pressure
1	0.913	0.32	22.31	69.718	456.16	0.64
1	10.818	0.172	24.629	143.19	747.36	0.66
1	0.759	0.277	31.574	113.99	1965.3	0.65
1	1.143	0.25	32.084	128.34	788.47	0.77
1	0.528	0.261	32.598	124.9	1384.2	0.7

of interest in the dataset.The performance of the generated models without shuffling is then compared with the performance of the models with shuffling. The features that show higher sensitivity to the shuffling operation are considered of higher importance to the model. Additionally, features with low importance are removed from the final training [26]. figure 3.2 illustrates the importance of the features used in this work for 5 different workers. It can be concluded from the figure that the feature with the highest importance score for all 5 workers except worker 4 was the ratio between the end-to-end distance and the length of the trajectory. On the other hand, the most important feature for worker 4 was the 20^{th} percentile of the pairwise velocity .

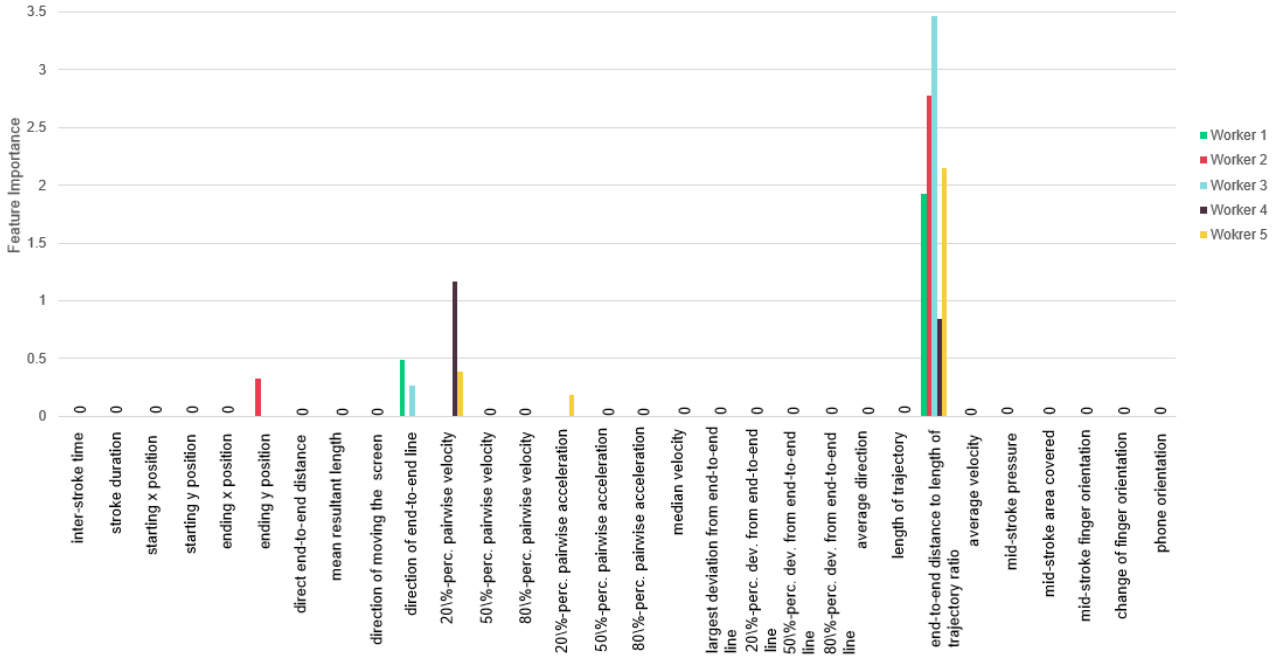


Figure 3.2: Feature Importance for 5 different workers

3.2 Model Generation

After preparing the data for training, supervised machine learning is used in order to train the models to distinguish between touch strokes that were generated by a genuine user and touch strokes that were generated by an impersonator. The supervised machine learning algorithms used in this work is Random Forest (RF). RF is an ensemble machine learning algorithm which relies on generating many decision trees and aggregating their results in the end by taking the majority vote. Every decision tree uses a different subset of features randomly selected from the original set of features and different training instances randomly selected, with replacement, from the entire training set. In this work we use a total number of 64 trees since as suggested by [27], this is threshold from which increasing the number of trees used in RF would bring no significant performance gain and would only increase the computational cost [27]. The main reasons behind choosing RF algorithm are summarized below:

1. It is less sensitive to outliers since it splits the data into groups based on a threshold value. This is important in this work, since some deviation from other data points may appear as the user is interacting with the touchscreen [28].

2. It automatically assesses the importance of the features when building the models [28].

3. It is accurate, stable and efficient for datasets with continuous and discrete form too since it makes a decision based on a particular threshold value [26].

4. It considers workers' possible behavioral changes over time since the final model aggregates a number of different temporal models [26].

All 40 workers' models that were trained using the data obtained from the first session are tested using data obtained from the second session. Three metrics are used for performance evaluation: Precision, recall and f1 score. The recall, as given in (3.1), is defined as the fraction of the genuine strokes predictions that are labeled correctly. On the other hand, the precision, as in (3.2) is defined as the number strokes of the

genuine user that are correctly labeled divided by the total number of the user's strokes. The f1 score combines both metrics using the harmonic mean as given in (3.3).

$$\text{Recall} = \frac{\text{no. of correct genuine strokes labels}}{\text{no. of correct genuine strokes labels} + \text{no. of incorrect nongenuine strokes labels}} \tag{3.1}$$

$$\text{Precision} = \frac{\text{no. of correct genuine strokes labels}}{\text{no. of correct genuine strokes labels} + \text{no. of incorrect genuine strokes labels}} \tag{3.2}$$

$$\text{F1score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{3.3}$$

As mentioned previously, multiple consecutive strokes should be combined together using a sliding window in order to obtain a better classification performance. To illustrate the effect of combining multiple strokes on the classification performance, Fig. 3.3 shows the f1 score average value of all classifiers over 10 runs for a varying number of strokes. As depicted in the figure , combining the strokes together improved the performance by 9%. Additionally, the performance started to converge towards 95% at n=7 strokes. Therefore, this was the number used to combine the strokes during training and testing. The performance of the models trained with random forest were compared with another powerful binary classifier: Support Vector Machine with an rbf kernel. SVMs are powerful classifiers which divide the feature space by a decision boundary such that the margin between classes is maximized. It uses kernel functions to solve classification problems where classes are not linearly separable by mapping the original feature space into another feature space where they are linearly separable [25]. Cross validation with five folds was used in order to perform hyperparameter tuning of the SVM models. Table 3.4 shows the average and the standard deviation of the precision, recall and f1 scores of the 40 workers considered in this work. It can be concluded from the table that, models trained with random forest perform better than those trained with SVM. Therefore, these models will be used at the recruitment stage in order to detect and eliminate the impersonators during the sensing task.

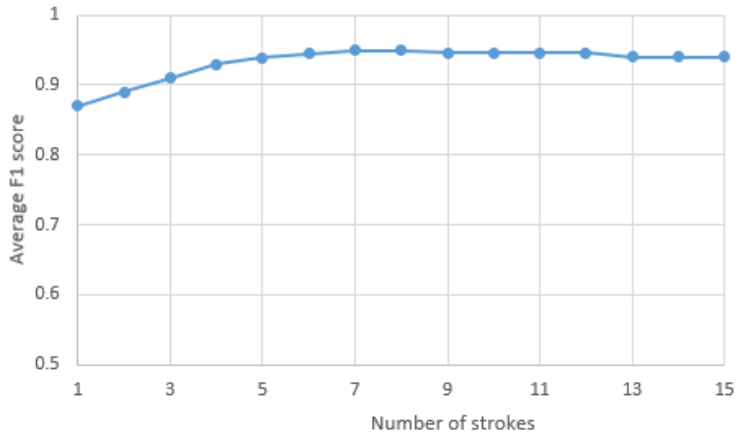


Figure 3.3: F1 score value averaged over 10 runs for a varying number of strokes.

Table 3.4: Evaluation of the machine learning algorithms using RF and SVM.

	Precision	Recall	F1 score
	Random Forest		
Average	0.956	0.951	0.954
σ	0.035	0.039	0.037
	Support Vector Machine		
Average	0.932	0.932	0.932
σ	0.064	0.058	0.062

CHAPTER 4

Touchscreen Input Behavioral Biometrics in Continuous MCS Recruitment

This chapter describes the overall proposed approach for integrating the trained behavioral biometrics models in continuous MCS recruitment systems. The different attributes used to represent a worker are first explained, then the recruitment mechanism adopted is discussed, taking into consideration the uncertainties accompanied with the predictions made by the model.

Typically, in continuous sensing tasks, readings are collected from participants' devices many times during the sensing period. Consequently, the recruitment systems adopted for one-time sensing tasks cannot be used for continuous sensing tasks, since it only considers the QoI of the group once, regardless of its value at any other point in time during the sensing period. In order to ensure that the publisher's required QoI is met during the sensing task, the recruitment system should continuously monitor the group's QoI and change the participants if the QoI was found to fall below the desired value. Therefore, every task is divided into equal shorter sensing intervals based on the publisher's requirements [3]. It is worth noting that behavioral biometrics can be used in

any dynamic recruitment system which keeps changing group members to maintain a certain QoI value. In this work, touch screen input behavioral biometrics is integrated with the dynamic recruitment system introduced in [3].

4.1 Parameter formulation and QoI Evaluation

In MCS systems, every worker can defined as $W_i =< l_i, SA_i, SF_i, RE_i, P_i, Conf_i >$. The different attributes used to define a worker are summarized in table 4.1. Using these attributes, different parameters that characterize a group of workers can be evaluated. These parameters can be classified into three main categories: AoI-related parameters, user-related parameters and device-related parameters [3].

Table 4.1: Different attributes used to define a worker

Symbol	Definition
l_i	*Location of worker* W_i
SA_i	*Set of sensors available in the device of worker* W_i
SF_i	*Sampling frequencies of the sensors available in the device of worker* W_i
RE_i	*Residual Energy of the worker* W_i
P_i	*Reputaion of worker* W_i
$Conf_i$	*Confidence that worker* W_i *is genuine based on the predictions made every interval*
B_i	*Behavioral Biometric model of worker* W_i

4.1.1 AoI-related Parameters

In order for higher quality sensing reports to be obtained from the group, it is not enough only to consider participants who are located within the AoI boundaries, however, it is also important to ensure that the participants cover the entire area of interest and are well distributed across it. Using the the GPS location of the workers in the AoI, two AoI related parameters can be evaluated : the coverage and the distribution. The coverage of a group $C(g)$ refers to the proportion of the sub-regions in AoI from which the required data is sensed. To evaluate the coverage of the group , first the AoI is divided into smaller sub-regions and then the coverage is found by dividing the number of sub-regions which include at least one group member (the number of covered sub-regions) over the total number of sub-regions in the AoI. Full coverage of the AoI is

achieved when participants submit sensing data from all sub-regions in the AoI. The coverage can be evaluated using (4.1) [3].

$$\mathrm{C(g)} = \frac{\text{no. of covered sub-regions in the AoI}}{\text{total no. of sub-regions in the AoI}} \tag{4.1}$$

On the other hand, the distribution parameter $D(g)$ measures how uniformly the participants are distributed in the AoI. It uses the Chi Square test in order to determine whether the observed values of the true number of users in each sub-region meet the the theoretical assumption that participants are evenly distributed among all sub-regions [3]. The following steps are followed in order to evaluate the distribution of the group:

1. Find the degrees of freedom

$$\text{degrees of freedom} = \text{number of sub-regions} - 1 \tag{4.2}$$

2. Find the alpha risk alpha risk= 1- test's confidence level

3. Find E_i which refers to the expected number of users in each sub-region.

$$E_i = \frac{\text{no. of group members}}{\text{no. of sub-regions}} \tag{4.3}$$

4. Find O_i for all sub-regions, which refers to the true number of users available in sub-region i

5. Evaluate the Chi Square test statistic (x^2) as follows:

$$x^2 = \sum_{i=1}^{\text{total number of subregions}} \frac{(O_i - E_i)^2}{E_i}; \text{for } E_i \neq 0 \tag{4.4}$$

6. From the table of probabilities of the Chi-Square distribution, find θ_U which is the value to which the accepted test statistic will be compared.

7. If $x^2 > \theta_U$, the assumption of even distribution is rejected and the distribution of the group is set to zero. If $x^2 \leq \theta_U$, then the assumption of the test statistic is

accepted and the distribution is evaluated as:

$$D(g) = 1 - \frac{x^2}{\theta_U}; \text{for } \theta_U \neq 0 \tag{4.5}$$

4.1.2 Device Related Parameters

The device related parameters reflect the capability of the participants' devices to sense the requested data. Two device-related parameters are used in the evaluation of the QoI of the group : the sampling frequency of the sensors in the group $SF(g)$ and the residual energy of the devices in the group $RE(g)$. The sampling frequency of a sensor reflects the rate at which the data can be sensed and reported. Having sensors with high sampling frequencies is important especially in continuous sensing tasks, since they require that the sensing data is reported immediately. To evaluate the $SF(g)$, first, the sampling frequency of each sensor s in the group is evaluated as given in (4.6), where n is the group size and $SF_i(s)$ is the sampling frequency for sensor s of participant i. In this equation, both the mean and the standard deviation are taken into consideration since the arithmetic mean on its own is not a fair measure. Once SF(g,s) is evaluated for all sensors, the overall sampling frequency of the group can be evaluated using (4.7) where SD refers to the standard deviation [3].

$$\text{SF(g,s)} = \frac{1}{n} \sum_{i \in g} SF_i(s) \times e^{-SD(SF_i(s))} \tag{4.6}$$

$$\text{SF(g)} = \sum_{s_j \in \text{ set of required sensors}} SF(g, s_j) \tag{4.7}$$

Since participants are continuously submitting sensing reports to the management platform, it is important to consider the drainage of the battery of their devices during the task and update the management platform about it. The battery drainage of each device can be either caused by the sensing task or by the device's brand, age and the number and type of the applications running on the device. Every sensing period, the residual

energy of each' device can be updated using (4.8) [3].

$$RE_{t+1} = RE_t - \text{total drainage} \tag{4.8}$$

The collective residual energy of a group of workers is evaluated by considering the mean and the standard deviation of the group members' residual energies, similar to the case of the sampling frequency. The group's residual energy can be evaluated as given in (4.9), where RE_i is the residual energy of participant i [3].

$$\text{RE(g)} = \frac{1}{n}\sum_{i \in g} RE_i \times e^{-SD(RE_i)} \tag{4.9}$$

4.1.3 User Related Parameters

User-related parameters reflect the properties of the users that can affect the sensing outcome.Two user-related parameters are considered in this work : the reputation of the group $(P(g))$ and the safety of the group $(S(g))$. Continuous sensing tasks require the commitment of the workers throughout the whole sensing period, however, not all workers are committed to completing the sensing task [3].Based on the workers' historical performance in previous tasks, the reputation can be evaluated as follows:

$$P_i = \frac{\text{number of tasks the worker was committed to}}{\text{total number of assigned tasks}} \tag{4.10}$$

Having a member with a low reputation value in the group can significantly affect the quality of the group's sensing, therefore, the minimum reputation value is used to represent the reputation of the group as a whole as given in (4.11) where $i \in g$ [3].

$$\text{P(g)} = min_{i \in g}\{P\} \tag{4.11}$$

Since continuous sensing tasks span over a period of time, the number of impersonators selected in the group can severely degrade the quality of information, since multiple sensing reports are collected from each worker and there is a high chance that the sub-

mitted sensing reports by the impersonators are not truthful. Fortunately, it is possible to detect impersonators in the group every sensing interval from the workers' unique way of interacting with their smartphones' touchscreens. Depending on the nature of the task, the task requester divides the total sensing period into equal intervals. Consequently, the system can predict whether or not the worker was genuine during a sensing interval using worker's previously built behavioral biometrics models and the touch stroke predictions made in previous sensing intervals. The confidence metric is a measure of trustworthiness that a user is genuine based on the the touchscreen input data obtained from previous sensing intervals. The confidence of a user can be evaluated using (4.12).

$$Conf_i = 1 - \frac{\text{duration the worker was impersonated during the task}}{\text{duration the worker was committed to the task}} \tag{4.12}$$

Since any member with a low confidence value can degrade the quality of sensing, the minimum value is used to evaluate the confidence of a candidate group, similar to the way the reputation of the group was found. The confidence of the group can be evaluated using (4.13).

$$\text{Conf(g)} = min_i\{Conf\} \tag{4.13}$$

The six previously introduced parameters can be used to find the QoI of the group as in (4.14), where $w_1 - w_6$ are the weights assigned to each of the parameters as specified by the task publisher [3].

$$QoI(g) = w_1 \times C(g) + w_2 \times D(g) + w_3 \times SF(g) + w_4 \times RE(g) + w_5 \times P(g) + w_6 \times Conf(g) \tag{4.14}$$

4.1.4 Stability

In continuous sensing tasks, some participants might leave the AoI or loose connectivity with the management platform during the sensing period. Since the sensing is required to be continuous, recruiting such participants is not desirable. Nevertheless,

it is possible to predict participants' future mobility patterns using their historical mobility traces, since people are very likely to visit the same places everyday. Using the predictions of the participants' locations over the required sensing period in the AoI, the coverage that is expected to be achieved at the beginning of each sensing interval C_i can be evaluated. The stability parameter reflects the availability of the participants in the AoI by the summing the expected coverage at every sensing interval during the sensing period, as given in (4.15) [3].

$$Stability = \sum_{i=1}^{\text{number of sensing intervals}} C_i \tag{4.15}$$

4.2 Recruitment Mechanism

Once a sensing task is publicized, the goal of the recruitment system is to find the group with the highest stability and confidence whose members are able to achieve the required QoI during the period of sensing. A summary of the recruitment system is illustrated in Fig. 4.1. The system starts initially by searching for a group of participants whose stability and confidence are maximized using the genetic algorithm. The confidence parameter was considered along with the stability in the initial recruitment step to reduce the chances of impersonation attacks in the first sensing interval. The pseudocode of the algorithm used is given in Algorithm 1. Starting with a randomly generated population, a fitness function is used to evaluate every group in the population. The fitness function used in the genetic algorithm is the sum of both, confidence and stability. Then, a set of groups are selected from the population using Roulette Wheel. The Roulette Wheel is set up such that groups with higher fitness values have higher probability of being selected. The selected groups from the Roullette Wheel represent the set of parents that will be used for the reproduction. After that, parents are combined together using crossover operation where a crossover point is chosen at random and participants are exchanged until the crossover point is reached. Next, groups generated from the crossover population are mutated by randomly replacing members in the group with another participants. Populations generated every iteration are referred

to as generations. The fitness values of the groups in each generation are evaluated and the process repeats until the maximum fitness fitness is achieved or a max number of iteration is reached or when the fitness value converges [29]. After the genetic algorithm selects the group with the highest stability and confidence, the QoI of the selected group is evaluated and new members are added to the group in case the obtained QoI value is less than the required value by the task publisher. Every sensing interval, the group members' behavioral biometric models are used to detect the impersonators in the group. After that, a trust value is evaluated to find the confidence in the predictions made by the machine learning model. Based on the obtained trust value, the system removes impersonators from the group and then the QoI value of the group is checked again [3].

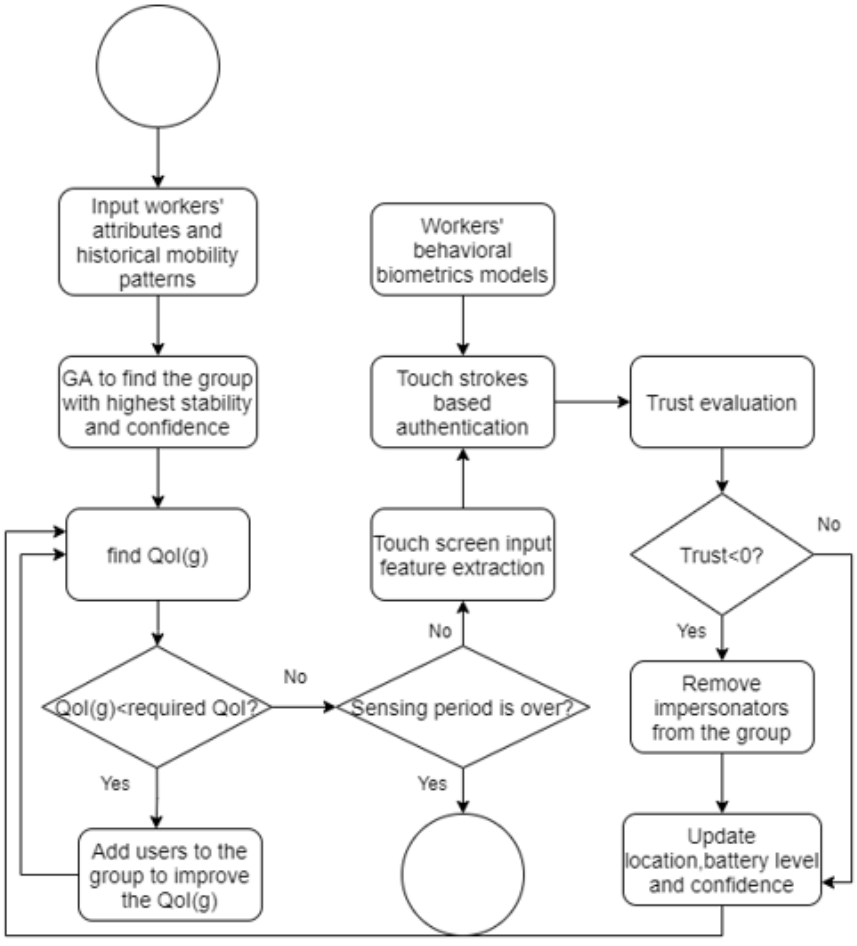


Figure 4.1: Flowchart of the proposed approach

Algorithm 1 : Genetic algorithm to select the group with highest stability and confidence

Input: participant's mobility predictions,individual confidence parameters, AoI boundaries, sensing period.
Output: a group that achieves the highest stability and confidence
Let G be the population of groups B = empty set; // B will hold the most stable groups of each size
G = empty list; // G will hold the population of groups
seed g = empty group; // seed g will hold the seed to the algorithm for each stage
for *group size=1* ***to*** *max group size* **do**
    best_g = empty group;
    best_fitness = 0;
    G=Initial population();
    if *seed g* $\neq \phi$ **then**
        G = G + seed g;
    end
    while *no fitness convergence and max fitness not achieved and max iterations not reached* **do**
        for *all* $g \in G$ **do**
            evaluate stability(g);
            evaluate confidence(g);
            fitness(g)=stability(g)+confidence(g);
        end
        current_population_best_ g= find_group_with_highest_fitness;
        if *current_population_best_g's fitness* $>$ *best_fitness* **then**
            best_g =current population_best_g;
            best_fitness=current_population_best_g's fitness;
        end
        P = empty list // P will hold the set of selected parents for reproduction
        C = empty list // C will hold the set of parents after performing crossover
        M = empty list // M will hold the set of parents after performing mutation
        P = Roulette wheel(G);
        C = one point crossover(P);
        M = substitution mutation(C);
        G = M;
    end
    B = $B \cup best\_g$;
    seed g=best_g + find_good_ member();
end
best_g=rank_fitness_return_ best(B); // find the group that achieves the highest fitness
return best_g;

4.2.1 Integrating Behavioral Biometrics in Continuous MCS Recruitment

After every sensing interval, the system uses participants' touch screen input and their behavioral biometrics models in order to predict whether or not each participant is genuine. However, the use of machine learning models is always accompanied by uncertainties regarding their outcomes. Since no guarantee is provided for the correctness of the predictions made by the models, the system needs to deal with inherent uncertainty in its outcome in order to make the used machine learning models more dependable [30]. Trusting a machine learning model can be interpreted as a special type of belief, where the model believes that users behaves in a certain way. In order to quantitatively evaluate the trustworthiness of a machine learning model, the uncertainty in the belief needs to be taken into consideration [31]. In the context of this work, three special cases can take place:

1. The machine learning model believes that the user is a genuine user without any uncertainties.

2. The machine learning model believes that the user is not genuine without any uncertainties.

3. The machine learning model is uncertain about whether or not the user is genuine.

The trust metric of a machine learning model can be evaluated using entropy as given in (4.16), where p is the probability that the user is genuine based on the machine learning model's predictions and $H(p)$ is the entropy function which represents the model's uncertainty. The trust metric gives a positive value in the first special case, a negative value in the second special case and 0 in the third special case [31]. Therefore, in our proposed approach, whenever the trust value is negative, the user is eliminated since the

model is confident that the user is an impersonator.

$$T = \begin{cases} 1 - H(p) & ,0.5 \leq p < 1 \\ H(p) - 1 & ,0 < p < 0.5 \\ 1 & ,p = 1 \\ 0 & ,p = 0 \end{cases} \tag{4.16}$$

The probability that the user is genuine can be evaluated as given in (4.17), by considering the classifier's probabilistic scores obtained for each genuine stroke prediction s_i and the fraction of the positively labeled strokes by the model n/N, where n is the number of positively labeled strokes and N is the total number of predictions made by the model.

$$p = \frac{n}{N} \sqrt{\prod_{i=1}^{n} s_i} \tag{4.17}$$

Using the evaluated probability, the entropy function can be evaluated as given in (4.18).

$$H(p) = -p \log_2(p) - (1-p) \log_2(1-p) \tag{4.18}$$

4.2.2 Feedback Mechanism for the Proposed Approach

Each sensing interval, the MCS system uses the touch screen input obtained from each user to detect and eliminate impersonators from the group. After a decision is made by the system to keep or remove each participant in the group, all confidence values are updated as given in (4.19) by taking the weighted sum of the previous confidence values and the new confidence value. In this equation, α is the weight of the new confidence value chosen by the task publisher.

$$conf_i = (1-\alpha) conf_i + \alpha (1 - \frac{\text{duration the worker was impersonated during the task}}{\text{duration the worker was committed to the task}}) \tag{4.19}$$

In return for the submitted sensing reports of the group each interval, the total payment provided to the group can be evaluated using(4.20) ,where the maximum QoI represents

the maximum QoI value that a group can achieve if all parameters were maximized and the maximum interval budget is the maximum budget assigned by the publisher for each interval.

$$\text{TP} = \frac{QoI(g)}{\text{max QoI}} \times \text{ max interval budget} \tag{4.20}$$

Every participant in the group is paid based on their contribution to the QoI(g). The contribution of a worker can be evaluated using (4.21), by finding the marginal QoI. Based on that, the payment is evaluated for each participant every sensing interval as given in (4.22), where $payment^{a}_{w_i}$ is the payment for worker w_i at interval a . At the end of the task, all participants are paid for all the intervals they participated in using (4.23),

$$\text{contribution of worker } w_i = QoI(g) - \text{QoI(group without worker } w_i) \tag{4.21}$$

$$payment^{a}_{w_i} = \frac{\text{contribution of worker } w_i \times TP}{QoI(g)} \tag{4.22}$$

$$TP_{w_i} = \sum_{a=1}^{\text{num of sensing intervals}} payment^{a}_{w_i} \tag{4.23}$$

CHAPTER 5

Simulation Results and Discussion

In this chapter the proposed approach is simulated in an untruthful environment to prove its robustness. The dataset used in order to obtain users' locations over a certain period of time is the vehicular mobility traces of the city of Cologne dataset [3]. In all experiments,the AoI boundaries were set to (10000 to 15000) x (10000 to 15000). The remaining user attributes were randomly generated following a uniform distribution. In addition, the previously trained behavioral biometrics models and touchstroke features were assigned for each user in the dataset at random. The performance of the proposed approach is compared with the stability-based group based recruitment system (stability-based GRS) proposed in [3]. The stability-based GRS is a dynamic recruitment system for continuous sensing tasks where the most stable group of participants are selected initially using the genetic algorithm and then during the sensing task, participants who leave the AoI are removed and new users are added to the group if the QoI(g) was found to drop below the required value. The system gives a higher emphasis to participants' mobility during the sensing period in order to ensure a certain QoI value. Therefore, to compare with our proposed approach, the selection mechanism is simulated using the same equations proposed in section 4.1.

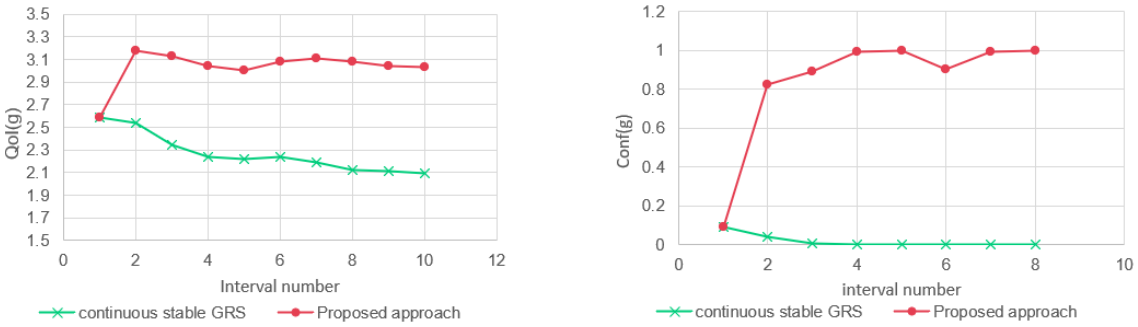


Figure 5.1: Achieved QoI and confidence during each sensing interval with 10% impersonators in the population.

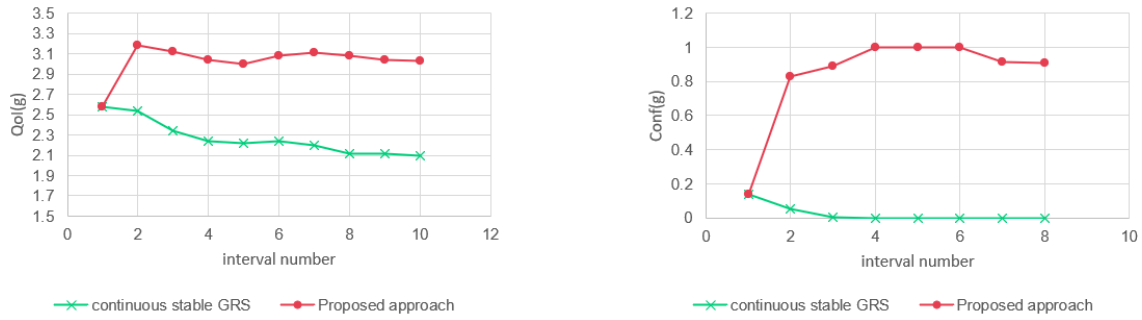


Figure 5.2: Achieved QoI and confidence during each sensing interval with 20% impersonators in the population.

5.1 Performance Evaluation in One Sensing task

In this experiment, a sensing task with 10 sensing intervals and a required QoI of 2.5 was simulated where the percentage of impersonators in the AoI was changed from 10% to 40%. The simulation results are shown in figures (5.1-5.4) where the figure s to the left show the QoI of the group each sensing interval and the figures to the right shows the confidence achieved by the group each sensing interval. As illustrated in the figures, it takes the proposed approach one interval only to meet the required QoI value and achieve a higher confidence value during the rest of the sensing task period, even when 40% of the population were impersonated. This is due to the fact that the system needs to wait to obtain touchscreen input data from the users during the first sensing interval. In addition, it is clear from the figures that the proposed approach performs better than the stability-based GRS since the stability-based GRS fails to achieve the required QoI during the sensing period due to the decreasing confidence value of the group.

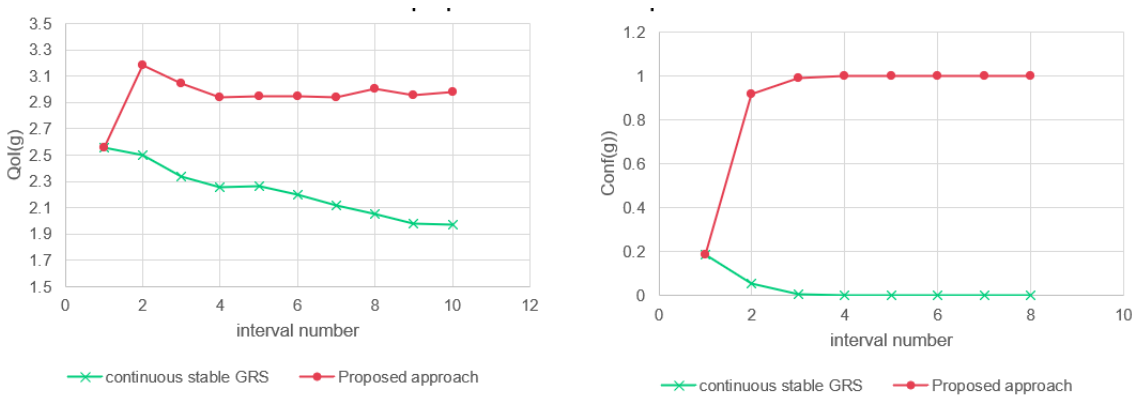


Figure 5.3: Achieved QoI and confidence during each sensing interval with 30% impersonators in the population.

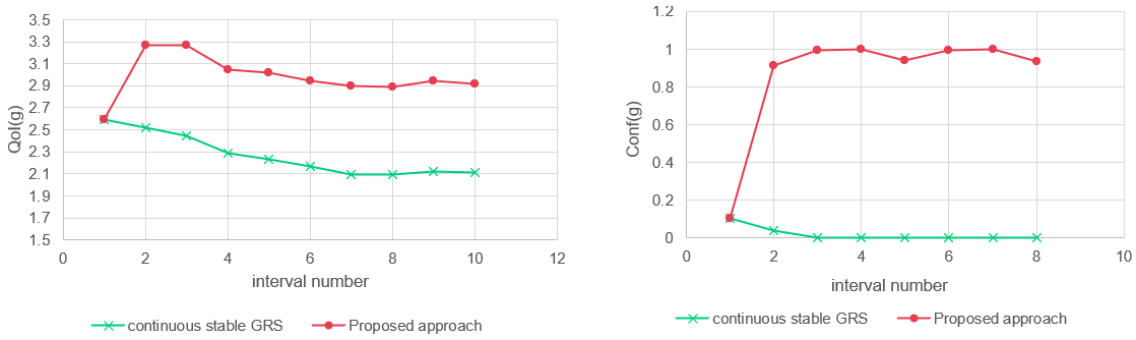


Figure 5.4: Achieved QoI and confidence during each sensing interval with 40% impersonators in the population.

5.2 Performance Evaluation in Multiple Sensing Tasks

A total of 100 sensing tasks with different number of sensing intervals and required QoI values were simulated in an AoI with 20% of its population being impersonated. To show how the proposed approach detects and eliminates impersonators, Fig. 5.5 shows the total number of impersonators selections found by summing all impersonators in the selected groups for every set of tasks averaged out over 10 runs. As shown in the figure ,the number of selections of impersonators made by the stability-based GRS can reach six time more than the number of selections made by the proposed approach. Therefore, the proposed approach outperforms the stability-based GRS in terms of not selecting impersonators throughout the sensing period. This is due to the fact that throughout the simulations, the impersonators' selections made by the proposed approach mostly would take place at the beginning of the sensing task, on the other hand, the stability-based GRS selects impersonators every interval and does not consider reducing their

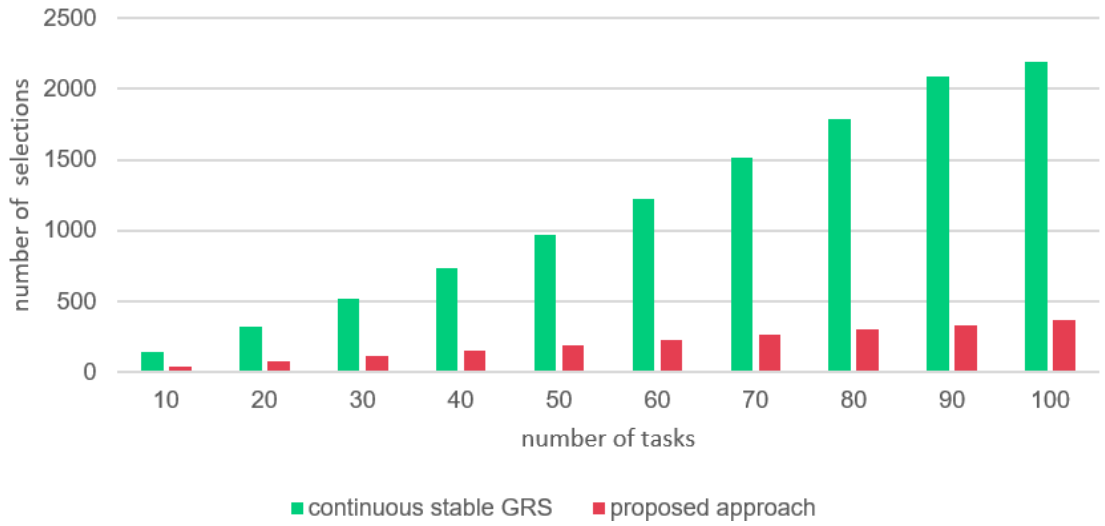


Figure 5.5: Number of selections of impersonators in the group for a different number of tasks

Figure 5.6: Performance evaluation of the proposed system

confidence which causes the number of impersonators selections to be bigger. Since the proposed approach uses machine learning in order to detect the impersonated workers, the performance of the system is evaluated using precision, recall and f1score given in (5.1) ,(5.2) and (5.3) every sensing interval, as shown in Fig. 5.6.

$$\text{Recall} = \frac{\text{no. of genuine users detected}}{\text{no. of correct genuine users detected} + \text{no. of incorrect non-genuine users detected}} \tag{5.1}$$

$$\text{Precision} = \frac{\text{no. of correct genuine users detected}}{\text{no. of correct genuine users detected} + \text{no. of incorrect genuine users detected}} \tag{5.2}$$

$$\text{F1score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{5.3}$$

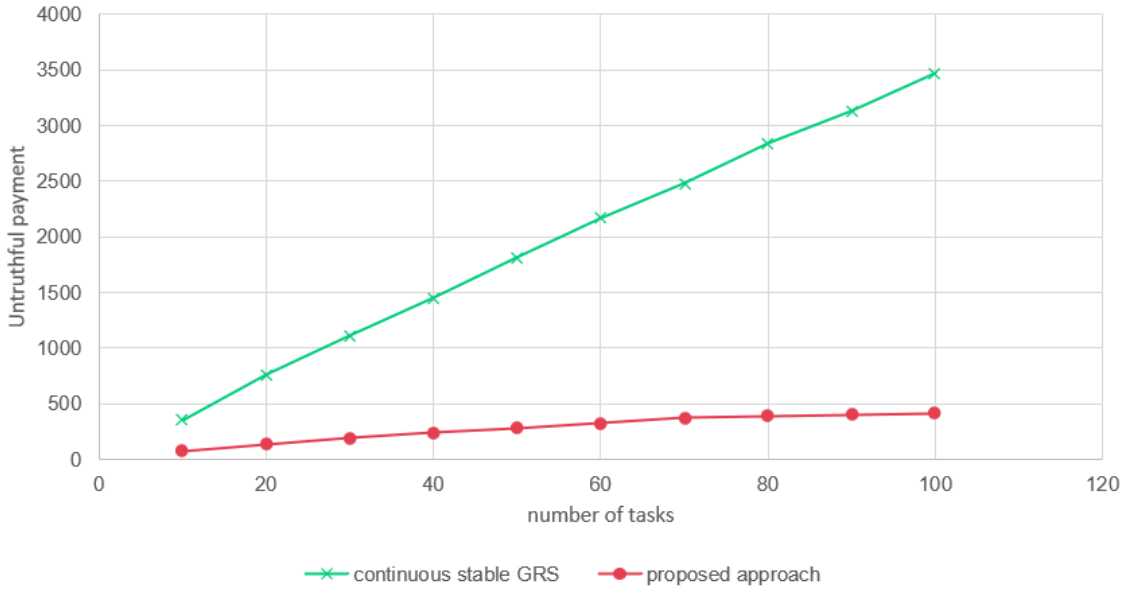


Figure 5.7: Untruthful payment

High scores for all metrics indicate that the system was able to accurately predict whether the user is a genuine user or an impersonators most of the times. Finally, the untruthful payment made using the proposed approach is compared to the untruth payment made using the stability-based GRS in Fig. 5.7. As expected, since the number of selections of impersonators made by the proposed approach is less than that made by the stability-based GRS, the untruthful payment made to the worker is also less.

CHAPTER 6

Conclusion

This chapter includes a summary of the final outcomes of the thesis and lists the challenges that could be faced by the proposed system.

6.1 Final Outcomes

In this thesis, touchscreen input behavioral biometrics was integrated with continuous MCS recruitment in order to detect and eliminate impersonators from the group every sensing interval. Two machine learning algorithms which are RF and SVM were implemented and evaluated to classify touch strokes made by a user into genuine strokes or non genuine strokes. Since the performance of the models built with RF outperformed the performance of the models built using SVM, RF was used during MCS recruitment to detect and eliminate impersonators after every sensing interval based on the data collected from users as they interacted with the smartphone's touchscreen. However, the adoption of machine learning may lead to some uncertainties in the predictions made by the model, therefore, a trust metric was proposed to help the system detect impersonators and genuine users in the system with higher confidence. Simulations were

performed in untruthful environments for one task and mulit-tasks to evaluate the performance of the proposed approach.The simulations performed in a sensing task with a specified required QoI showed that the proposed approach was able to maintain the required QoI value whenever it received touchscreen input from the users. This was achieved even when the percentage of impersonators in the group would reach 40% of the entire population. On the other hand, the simulations performed for multiple tasks showed that the proposed approach makes less number of selections of impersonators and consequently, decreased the untruthful payment significantly.

6.2 Limitations and future work

Even though using touchscreen behavioral biometrics in MCS recruitment was shown to be effective, the system still faces some challenges that need to be addressed. The different challenges faced by our system are listed below:

1. **Vulnerability to malware attacks**: In this work, it is assumed that impersonators have different touching behaviors from genuine users and therefore, the system was able to detect and eliminate them. However, some attackers may try to mimic the touching behavior of a user by placing a malware software on a worker's device which can read and report back the touchscreen input data of that user [25].

2. **New worker joining the system**: New workers joining the system have no data that can be used to build behavioral biometrics models for them. Therefore, the system will not be ableto detect whether these workers are impersonators or not during the sensing task [32].

3. **Behavior change and model retraining**: It is not enough to train the behavioral biometrics model of a worker once, since the behavior of a worker could change with time. Therefore, periodic retraining of the the models is necessary in order to avoid having low prediction accuracies. The main challenge here lies in deciding

when is the right time to retrain the model, while still considering the trade-off between the computational cost and accuracy [32].

For future work, touchscreen input can be combined with other types of behavioral biometrics such as: walking gait behavioral biometrics or keystroke dynamics behavioral biometrics in MCS recruitment in order to obtain an improved accuracy.

Bibliography

[1] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A Survey on Mobile Crowdsensing Systems: Challenges, Solutions, and Opportunities," IEEE Communications Surveys and Tutorials, vol. 21, no. 3, pp. 2419–2465, 2019, doi: 10.1109/COMST.2019.2914030.

[2] Y. Ren, W. Liu, Y. Liu, N. N. Xiong, A. Liu, and X. Liu, "An effective crowdsourcing data reporting scheme to compose cloud-based services in mobile robotic systems," IEEE Access, vol. 6, pp. 54683–54700, Aug. 2018, doi: 10.1109/ACCESS.2018.2868250.

[3] R. Azzam, R. Mizouni, H. Otrok, S. Singh, and A. Ouali, "A stability-based group recruitment system for continuous mobile crowd sensing," Computer Communications, vol. 119, pp. 1–14, Apr. 2018, doi: 10.1016/j.comcom.2018.01.012.

[4] M. El-Hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, "A survey of internet of things (IoT) authentication schemes," Sensors (Switzerland), vol. 19, no. 5, Mar. 2019, doi: 10.3390/s19051141.

[5] C. Miao, Q. Li, H. Xiao, W. Jiang, M. Huai, and L. Su, "Towards data poisoning attacks in crowd sensing systems," Proceedings of the International Symposium

on Mobile Ad Hoc Networking and Computing (MobiHoc), pp. 111–120, 2018, doi: 10.1145/3209582.3209594.

[6] K. Banti, F. Katsimpoura, M. Louta, and G. T. Karetsos, "Data quality in mobile crowd sensing systems: Challenges and perspectives," 2018 9th International Conference on Information, Intelligence, Systems and Applications, IISA 2018, 2019, doi: 10.1109/IISA.2018.8633627.

[7] L. Gonzalez-manzano, J. M. de Fuentes, A. R. L. User-related, L. Gonzalez-manzano, and J. M. de Fuentes, "Leveraging user-related Internet of Things for continuous authentication: a surveyd," vol. 53, no. June, 2019.

[8] Y. Liang, S. Samtani, B. Guo, and Z. Yu, "Behavioral Biometrics for Continuous Authentication in the Internet-of-Things Era: An Artificial Intelligence Perspective," IEEE Internet of Things Journal, vol. 7, no. 9, pp. 9128–9143, 2020, doi: 10.1109/JIOT.2020.3004077.

[9] W. Feng, Z. Yan, H. Zhang, K. Zeng, Y. Xiao, and Y. T. Hou, "A survey on security, privacy, and trust in mobile crowdsourcing," IEEE Internet of Things Journal, vol. 5, no. 4, pp. 2971–2992, 2018, doi: 10.1109/JIOT.2017.2765699.

[10] R. Estrada, R. Mizouni, H. Otrok, A. Ouali, and J. Bentahar, "A crowd-sensing framework for allocation of time-constrained and location-based tasks," IEEE Transactions on Services Computing, vol. 13, no. 5, pp. 769–785, 2020, doi: 10.1109/TSC.2017.2725835.

[11] C. Wu, T. Luo, F. Wu, and G. Chen, "EndorTrust: An endorsement-based reputation system for trustworthy and heterogeneous crowdsourcing," 2015 IEEE Global Communications Conference, GLOBECOM 2015, pp. 0–5, 2015, doi: 10.1109/GLOCOM.2014.7417352.

[12] Y. Gao, X. Li, J. Li, and Y. Gao, "DTRF: A dynamic-trust-based recruitment framework for Mobile Crowd Sensing system," Proceedings of the IM 2017 - 2017 IFIP/IEEE International Symposium on Integrated Network and Service Management, pp. 632–635, 2017, doi: 10.23919/INM.2017.7987347.

[13] X. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, "ARTSense: Anonymous reputation and trust in participatory sensing," Proceedings - IEEE INFOCOM,

pp. 2517–2525, 2013, doi: 10.1109/INFCOM.2013.6567058.

[14] D. Tao, P. Ma, and M. S. Obaidat, "Anonymous identity authentication mechanism for hybrid architecture in mobile crowd sensing networks," International Journal of Communication Systems, vol. 32, no. 14, pp. 1–16, 2019, doi: 10.1002/dac.4099.

[15] K. S. Killourhy and R. A. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," Proceedings of the International Conference on Dependable Systems and Networks, pp. 125–134, 2009, doi: 10.1109/DSN.2009.5270346.

[16] A. Buriro, B. Crispo, F. del Frari, and K. Wrona, "Touchstroke: Smartphone user authentication based on touch-typing biometrics," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9281, pp. 27–34, 2015, doi: 10.1007/978-3-319-23222-5_4.

[17] P. S. Teh, A. B. J. Teoh, and S. Yue, "A survey of keystroke dynamics biometrics," The Scientific World Journal, vol. 2013, 2013, doi: 10.1155/2013/408280.

[18] A. Nambiar, A. Bernardino, and J. C. Nascimento, "Gait-based Person Re-identification: A Survey," vol. 52, no. 2, 2019.

[19] A. I. Middya, S. Roy, S. Mandal, and R. Talukdar, "Privacy protected user identification using deep learning for smartphone-based participatory sensing applications," Neural Computing and Applications, vol. 6, 2021, doi: 10.1007/s00521-021-06319-6.

[20] C. Shi, J. Liu, H. Liu, and Y. Chen, "Smart User authentication through actuation of daily activities leveraging wifi-enabled IoT," Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), vol. Part F1291, 2017, doi: 10.1145/3084041.3084061.

[21] V. L. L. Thing, "IEEE 802.11network anamoly detection and attack Classification: A Deep Learning Approach," 2017 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6, 2017.

[22] L. Lu and Y. Liu, "Safeguard: User Reauthentication on Smartphones via Behavioral Biometrics," IEEE Transactions on Computational Social Systems, vol. 2, no. 3, pp. 53–64, 2015, doi: 10.1109/TCSS.2016.2517648.

[23] A. Serwadda, V. v. Phoha, and Z. Wang, "Which verifiers work?: A bench-

mark evaluation of touch-based authentication algorithms," IEEE 6th International Conference on Biometrics: Theory, Applications and Systems, BTAS 2013, 2013, doi: 10.1109/BTAS.2013.6712758.

[24] H. Xu, Y. Zhou, and M. R. Lyu, "Towards Continuous and Passive Authentication via Touch Biometrics: An Experimental Study on Smartphones," pp. 187–198, 2014.

[25] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," IEEE Transactions on Information Forensics and Security, vol. 8, no. 1, pp. 136–148, 2013, doi: 10.1109/TIFS.2012.2225048.

[26] M. Abououf, S. Singh, H. Otrok, R. Mizouni and E. Damiani, "Machine Learning in Mobile Crowd Sourcing: A Behavior-based Recruitment Model," ACM Transactions on Internet Technology (TOIT), accepted Feb 2021.

[27] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, "How many trees in a random forest?," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2012, vol. 7376 LNAI, pp. 154–168. doi: 10.1007/978-3-642-31537-4_13.

[28] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random Forests and Decision Trees," International Journal of Computer Science Issues (IJCSI) ,2012. [Online]. Available: www.IJCSI.org

[29] Rana Azzam, Rabeb Mizouni, Hadi Otrok, Anis Ouali, and Shakti Singh. Grs: A group-based recruitment system for mobile crowd sensing. Journal of Network and Computer Applications, 72:38–50, 2016.

[30] M. Kläs and A. M. Vollmer, "Uncertainty in Machine Learning Applications A Practice-Driven Classification of Uncertainty," 2018. [Online]. Available: https://www.kaggle.com/compet

[31] Y. L. Sun, Z. Han, W. Yu, and K. J. R. Liu, "A Trust Evaluation Framework in Distributed Networks: Vulnerability Analysis and Defense Against Attacks."25TH IEEE International Conference on Computer Communications. IEEE, 2006.

[32] M. Abououf, H. Otrok, R. Mizouni, S. Singh, and E. Damiani, "How Artificial

Intelligence and Mobile Crowd Sourcing are Inextricably Intertwined," IEEE Network, vol. 35, no. 3, pp. 252–258, May 2021, doi: 10.1109/MNET.011.2000516.