Real-Time Operating Hand Detection for the Optimization of Mobile Web Interfaces

Carla Fernández[1], José Flores[2], Brian Zylich[3], Martín González-Rodríguez[1], Miguel Labrador[4], Daniel Fernández Lanvin[1], and Javier De Andrés[5]

[1]Dept. of Computer Science, Universidad de Oviedo, uo244965@uniovi.es, martin@uniovi.es, dflanvin@uniovi.es
[2]Dept. of Electrical & Computer Engineering, Universidad del Turabo, jflores152@email.suagm.edu
[3]Dept. of Computer Science, Worcester Polytechnic Institute, bzylich@wpi.edu
[4]Dept. of Computer Science, University of South Florida, labrador@cse.usf.edu
[5]Dept. of Accounting, Universidad de Oviedo, jdandres@uniovi.es

***Abstract*— Mobile devices now rival desktop computers as a method of web browsing. Even so, many web applications are still designed with the desktop setting in mind. As screen sizes of mobile devices continue to get larger, operating smartphones single-handedly becomes increasingly difficult. This paper explores the possibility of automatic operating hand detection by capturing users' touch traces during normal browsing activities. Automatic operating hand detection would enable web applications to adapt their interfaces and better suit the user's laterality. Supervised classifiers were constructed for the primary goal of operating hand detection (left, right), but also to detect hand posture (thumb, index). Both classifiers use features extracted from the touch traces and button clicks on a dataset of 174 users. The resulting classifiers featured true positive rates of 96.0% and 70.1% respectively when tested using 10-fold cross-validation. While previous studies achieved similarly accurate results for operating hand (94.1%) and posture (82.6%) detection, the approach proposed by this paper is not platform-specific and does not rely on access to gyroscopes or accelerometers.**

***Index Terms*— laterality, posture, smartphone, touchscreen, web development**

I. INTRODUCTION

As of 2016, more than three quarters of Americans owned smartphones [1]. Mobile devices are increasingly rivaling desktop computers as the main form of interaction with online applications [2]. In the United States, one in ten adults uses a smartphone as their only method of online access [3]. Today, as the number of people browsing the Internet with mobile devices continues to rise, researchers have begun investigating smart mobile interface adjustment.

First-time users of a mobile application or website expect it to be easy to navigate with an intuitive design [4]. For some users, this first impression will characterize their perception of the entity represented by the mobile interface. In fact, 79% of consumers who have a bad navigation experience are more likely to seek out similar services from competitors [4]. Therefore, it is important for companies to focus on the optimization of design for ease-of-use and functionality for all users when designing a mobile interface.

There are external factors that greatly influence the user's experience navigating mobile applications and websites. Some of these factors are screen size, hand posture, and operating hand [5]. As the screen size of mobile devices increases, it becomes increasingly difficult to operate these devices with one hand ([2], [6]).

As can be seen in Figure 1, some mobile users with larger devices are unable to reach a large part of the screen when operating the device singe-handedly. One of the main issues arises when users reach for the top of the screen or the side opposite their operating hand. The figure shows an example interface and a side-by-side comparison of users attempting to navigate the website using their left and right hands. For the right-handed user, it is very hard to reach the navigation drawer located in the upper-left corner of the screen, circled in green. Meanwhile, the left-handed user is able to reach the area circled in green more easily, but has difficulty reaching the area circled in red.

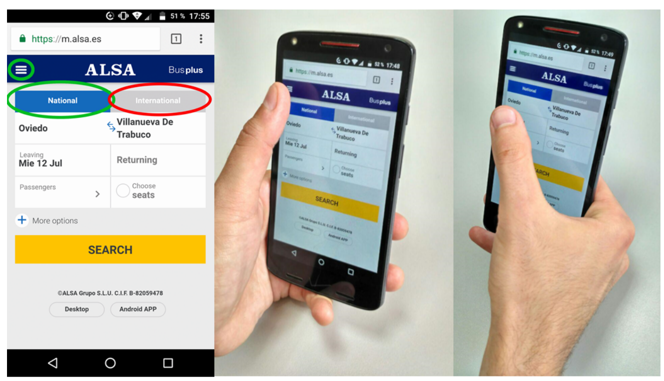

Fig. 1. On the left is an example of an interface that is not optimized for one-handed use. On the right is a comparison of thumb-based interaction for left-handed and right-handed users.

In general, laterality refers to the preference that people commonly exhibit for one side of their body over the other. Here, the term laterality refers to a user's preferred hand when operating a mobile device. The customization of mobile web interfaces to accommodate laterality has not yet been explored.

One way to allow users to comfortably interact with an application is through the provision of specialized interfaces that are tailored to provide a better experience for a certain population [7]. However, in most current applications these interfaces must be found and activated manually [6]. An automatic operating hand detection system would avoid these issues and allow developers to dynamically adapt the interface to better suit the user.

This paper explores the possibility of classifying users primarily according to their laterality, and secondarily according to their hand posture, by observing the way they interact with a mobile web application and limiting the use of resources to the information that can be gathered from any mobile device with a touchscreen.

Choosing not to require access to specific sensors avoids limiting the applicability of the solution for devices without these sensors or complicating the implementation of the solution into real web applications. Each device has different calibration values for the same sensors [8], which means that a cross-platform application, such as a mobile website, must account for device-specific bias and noise factors that would impose a higher burden on web developers interested in applying this research.

The rest of the paper is organized as follows: Section II details the state of the art in specialized ergonomic interfaces and operating hand detection. Section III describes the approach used in this research. Section IV compares the proposed algorithms with current alternatives. Finally, Section V concludes the paper.

II. RELATED WORK

Previous works have explored some of the aspects related with this research area.

A. *Specialized ergonomic interfaces*

A study done by Zheng et al. [9] explores the adaptation of mobile user interfaces according to various aspects of a user's environment including location, time, and ambient conditions. Another system developed by González-Rodríguez et al. [7] collected data such as age, visual accuracy, physical disabilities, and interaction preferences in order to automatically customize communication and interaction channels. Each of these systems allows for the customization of applications' interfaces based on various parameters specific to the user and their environment. However, the user's operating hand, one of the most important factors in the usability of mobile applications today, was not considered.

In the research done by Guo et al. [2], interface adjustment according to operating hand was briefly explored. The study recognizes that many features should be reorganized to better suit mobile users depending on their laterality. However, they also observed that each application has buttons and methods of input that are often very different in significance and function, concluding that the adjustment of the interface is best left to the software engineers designing the applications.

B. *Operating Hand and Posture Detection*

Various studies have used data gathered from the touchscreen as a part of their operating hand and posture classifiers. Some used this data in conjunction with accelerometer and device orientation readings ([6], [10], [11]).

One type of data gathered from the touchscreen is the change in touch size depending on screen location. GripSense [10] and Löchtefeld et al. [6] use this data and the assumption that more contact will be made on the side of the phone further away from the thumb and less on the side closest to the thumb to help detect operating hand. Additionally, GripSense [10], Löchtefeld et al. [6], and Seipp et al. [11] determined that the contact area is much larger when using the thumb than when using the index finger. In each study, an application was developed that forced the test subjects to swipe or click on specific areas of the screen, therefore gathering the necessary information to distinguish between operating hand postures. However, all three systems relied heavily on data from gyroscopes in their classification process and all three were specific to the Android platform.

Another type of touch data used in operating hand classification systems is touch position. Seipp et al. [11] found the X-offset of a button touch to be a strong indicator of operating hand. In a study by Löchtefeld et al. [6], a swipe-to-unlock application was developed to determine the user's operating hand. This application found touch position to be a key feature, as users operating the device with their right thumb tended to swipe from the center to the right, whereas those using their left thumb would usually swipe to the left. Although this study achieved high rates of accuracy, their solution is specific to the phone-unlocking process, an activity that is not commonly used while navigating through web applications. This same issue was found in a paper by Buriro et al. [12].

One final technique for the interpretation of touch data involves extracting features from longer touch traces. GripSense [10] incorporated touch trace analysis as a factor in its approach. The system used the heuristic that thumb-based traces generally form a much more pronounced arc than when using an index finger. This heuristic was incorporated by biasing toward a thumb-based interaction if the X-displacement was greater than 5% of the screen resolution. In 2016, Guo et al. [2] developed a system to classify a user's operating hand with only touch data. Their system analyzed features such as trace length, velocity, X and Y displacements, curvature, and convex orientation. However, this study is specific to the Android platform and was not applied to web applications.

Based on the aforementioned studies, this research was conducted to test the validity of three main hypotheses:

- **H1:** Laterality detection concepts featured in previous studies can be applied to the mobile web domain. All prior research examined laterality detection using Android applications.
- **H2:** Laterality detection using machine learning can achieve high accuracy while collecting data in a public, uncontrolled environment. Studies, such as [2], achieved highly accurate results but were conducted in a controlled environment where subjects were required to swipe in a specified manner. Our goal is to eliminate the behavioral bias caused by unnatural conditions described by Kaikkonen et al. [13].

- **H3:** Laterality can be accurately detected using only data gathered from the touchscreen during normal browsing activities (e.g. button clicks and scrolling) and without the use of additional sensors.

III. Methodology

This approach is based on the construction of supervised classifiers to predict a user's operating hand and posture based on touch traces generated through normal operation of the mobile device. First, touch data was collected using a series of tests located in a web application. Second, the data was filtered to remove incomplete data and separate each touch trace into a set of subtraces. This step was necessary because a single trace could be the result of a user swiping up and down multiple times without lifting their finger. Therefore, every time the trace changed direction, a new subtrace was created to preserve the validity of features such as start points, slope, and maximum and minimum X-positions. Next, each subtrace was passed through a secondary filter to ensure it contained enough data points to provide meaningful results. After this preprocessing stage, the feature vectors detailed in Table I were computed for each user. Lastly, supervised classifiers were constructed that output predictions of users' operating hand and posture. Operating hand is classified as (*a*) left or (*b*) right, while posture is classified as (*a*) thumb or (*b*) index.

A. Data Acquisition

To simulate a mobile web environment, a web application was developed containing a series of tests. To avoid the use of external libraries, the native TouchEvent API for Javascript was used. This API allows developers to listen and respond when a user initiates or finishes a touch trace and periodically throughout each touch trace.

Two main categories of data were collected: data relating to button clicks and data relating to the user's swiping behavior. The web application designed was composed of seven pages. Out of these, four pages had buttons that spanned the width of the page and facilitated the forward navigation through the activity. The position where the user clicked on each button was recorded, measuring the X and Y position of each click. Additionally, two tests were conducted to collect scroll data: the first consisted of finding an object at the bottom of a vertical panorama and the second involved finding an uppercase 'A' within a body of text. Both tests were specially designed so that mobile users were required to scroll up and down. Scroll data was recorded as a collection of points, each with X and Y coordinates. The time elapsed during each scroll was also recorded. At the end of the experiment, a form asked the user to indicate the operating hand and posture used during the tests and also provide information for use as control variables such as: gender, age, weekly computer usage (hours/week), and device type. Lastly, the application automatically stored the user's screen width.

B. Construction of the Feature Vectors

Each classifier was constructed from slightly different feature sets, as seen in Table I. The chosen features were based on previous studies ([2], [11], [10]) and refined for this project's dataset and purpose.

TABLE I
FEATURES USED IN HAND AND POSTURE CLASSIFIERS

Feature	Hand	Posture
screen width	X	X
average x (click)	X	
median x (click)	X	X
median y (click)		X
variance x (click)		X
variance y (click)		X
standard deviation y (click)		X
average slope (scroll)	X	X
average x (scroll)	X	
median x (scroll)		X
average start x (scroll)	X	
average maximum x (scroll)	X	
average minimum x (scroll)	X	
standard deviation x (scroll)	X	X
standard deviation y (scroll)	X	X
average y displacement (scroll)		X

1) Click X and Y Positions: The statistical average, median, variance, and standard deviation were calculated for the set of X coordinates and the set of Y coordinates from each of the user's clicks. Each X and Y coordinate is relative to the button being clicked. All of these values were initially calculated. Table I contains only those features that were found to be beneficial after testing.

2) Average Slope: Average slope for a user's touch traces was calculated by finding the slope from the start point to the end point (Formula 1) of each trace and averaging them.

$$m = \frac{\Delta y}{\Delta x} = \frac{y_{n-1} - y_0}{x_{n-1} - x_0} \tag{1}$$

3) Scroll X and Y Positions: For each user, the position of each point recorded along their scrolls was examined. From this data, the user's overall average and median X-value were determined. The maximum, minimum, and initial X-values were found for each of a user's touch traces and later averaged. Lastly, the standard deviation was calculated for the user's set of X-values and Y-values from all of their scrolls.

4) Average Y Displacement: The displacement from the maximum y-value along each scroll to the minimum y-value was calculated (Formula 2) and then averaged over the set of scrolls for each user.

$$|\Delta Y| = Max(Y) - Min(Y) \tag{2}$$

IV. Results & Performance Comparison

One of the issues found in previous studies on the detection of operating hand is that they have a low number of test subjects: 14 [2], 32 [5], 12 [6], 14 [11], 10 [10]. A major goal in this study was to gather data from a large number of users to better simulate the data that would be gathered from a real web application with many diverse users.

The data gathering phase consisted of a 3-day public stage, in which the site was made public[1], yielding data from 174 users. The sample was composed of professors, family, and friends who were easily reachable through social media. For this reason, the most common age groups match university students and their close relatives.

Of the users that participated in this study, 53% used their right thumb, 27% used their right index finger, 16% used their left thumb, and 4% used their left index finger. Therefore, the dataset contained a percentage of left-handed users that doubled the global left-handed percentage (approximately 10% [14]). This was, however, beneficial for this project, since it provided more left-handed data for the machine learning algorithms.

Given the age distribution shown in Figure 2, conclusions cannot be drawn from this study regarding the profiling of children and elderly users. Furthermore, 44% of the test subjects were female, compared to the approximately 50% found in the global population [23]. These limitations should be noted when considering the results.

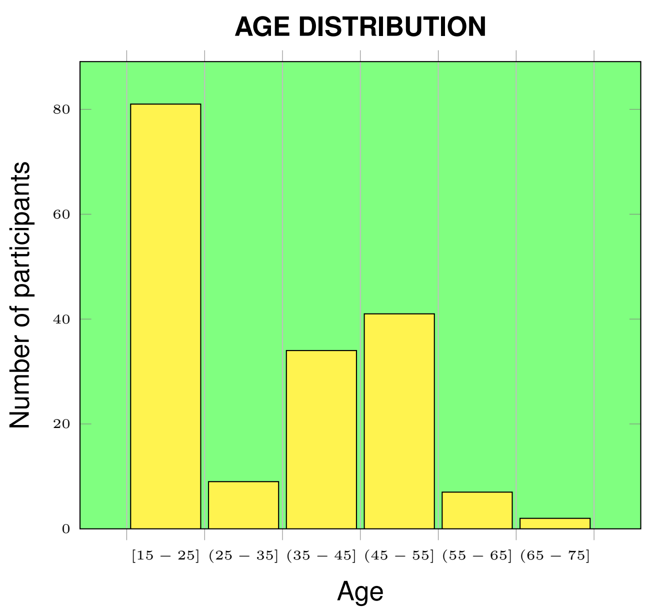

Fig. 2. Distribution of the ages of participants in this study.

A. Performance Comparison

To analyze the data, six classifiers were constructed and compared to evaluate their performance. The constructed classifiers were: Random Forest (RF) [15], Naive Bayes (NB) ([16]), *Weka*'s J48 [17], k-Nearest Neighbors (KNN) [18], Sequential Minimal Optimization (SMO) [19], and Adaptive Boosting (AB) [20]. Some of these algorithms (RF, NB, J48, KNN) were chosen because of their use in previous studies ([2], [6], [11], [10]). The remaining algorithms were selected for comparison because they were found to be particularly successful using the data collected in this study.

In this paper, True Positive Rate (TPR), F-Measure (F1), and the Matthews Correlation Coefficient (MCC) [21], were used to evaluate performance.

TABLE II
COMPARISON OF OPERATING HAND CLASSIFIERS

	TPR	F1	MCC
RF	95.4%	95.4%	0.854
NB	93.1%	93.2%	0.790
J48	91.4%	91.4%	0.735
KNN	92.5%	92.5%	0.765
SMO	93.7%	93.6%	0.798
AB	96.0%	96.0%	0.874

The Operating Hand Classifier was tested by ten-fold cross-validation [22] on the set of 174 users. Using Adaptive Boosting, 136 (97.84%) out of 139 right-handed and 31 (88.57%) out of 35 left-handed users were correctly classified. As a result, and comparing the measures found in Table II, it was determined that the Adaptive Boosting algorithm achieved the best performance for the classification of operating hand. This result supports hypotheses H1, H2 and H3, showing that a system to automatically detect the user's operating hand can be designed for mobile web applications using only touchscreen data gathered in an uncontrolled environment.

For comparison, Guo et al. [2] achieved a TPR of 94.1%, also analyzing data gathered from the touch screen. In another study, Löchtefeld et al. [6] attained a TPR of 98.51%. Although their system is slightly more accurate than the system described in this paper, they focus on the phone-unlocking process and use sensors, limiting its applicability to a web environment.

For the comparison of the Hand Posture Classifiers, False Positive Rate (FPR) was included to evaluate performance. As 69% of the users in the dataset used thumbs and only 31% used their index fingers, FPR was added as a metric for comparison in order to prevent choosing a classifier that took advantage of overfitting to achieve a high TPR.

TABLE III
COMPARISON OF HAND POSTURE CLASSIFIERS

	TPR	FPR	F1	MCC
RF	70.1%	58.3%	64.6%	0.180
NB	70.1%	59.3%	64.0%	0.173
J48	67.8%	61.3%	61.9%	0.098
KNN	70.7%	59.0%	64.5%	0.192
SMO	72.4%	61.3%	63.7%	0.282
AB	70.7%	62.1%	62.5%	0.180

Ten-fold cross-validation on the set of 174 users was also used to test the Hand Posture Classifiers. As can be seen in Table III, SMO had the highest TPR and MCC. However, SMO also had a higher FPR and a lower F-Measure than other classifiers, indicating a strong bias for one classifi-

[1] https://lateral.herokuapp.com/en

cation over the other. Similarly, Adaptive Boosting and k-Nearest Neighbors achieved higher TPR's but exhibited more overfitting as evident by their FPR's and MCC's. Therefore, Random Forest was determined to be the best algorithm for the classification of hand posture, as it had the lowest FPR and highest F-Measure, indicating that it was least influenced by overfitting. Using Random Forest, 112 (93.3%) of the 120 thumb-interaction instances were correctly classified, while 10 (18.15%) of the 54 index-interaction instances were correctly classified.

In contrast, Seipp et al. [11] were able to distinguish between users operating the device with their index finger and thumb with a TPR of 82.6%. However, the data collected in their study was from one device running an Android-specific application that relied on sensor readings from the phone's gyroscope and accelerometer.

Furthermore, for the purpose of mobile interface adjustment, it is better to have a higher FPR that favors thumb classification than index classification. This is because users operating a device with their index finger will not be adversely affected by interface optimization for thumb-based interaction.

V. CONCLUSIONS & FUTURE RESEARCH

The main goal of this research was to determine the operating hand of a mobile device user in a web application. Analyzing button clicks and touch traces generated by scrolling, our solution allowed for very accurate detection of whether the user was operating the device with their left or right hand. To the best of our knowledge, this is the first paper to explore and propose a solution for operating hand detection in mobile web applications using only data gathered from the touchscreen.

Companies and organizations will be able to improve the user experience of their mobile web sites by providing custom interfaces for left and right-handed users.

This research will allow them to automatically recommend the corresponding interface to the user.

In order to improve hand posture classification, more data should be gathered from left-handed users, especially while operating the device using the index finger. Additionally, measuring the touch size and pressure of the finger on the screen might provide a better distinction between index finger and thumb-based interaction.

One topic that warrants additional exploration is the use of data gathered from web browsing behaviors to determine whether a user of a mobile website is a child, an adult, or an elderly person.

ACKNOWLEDGMENTS

This research has been partially supported by the National Science Foundation under grants No. 1458928 and No. 1645025, An REU Site on Ubiquitous Sensing, and by the European Union, through the European Regional Development Funds (ERDF), and the Principality of Asturias, through its Science, Technology and Innovation Plan (grants GRUPIN14-100 and GRUPIN 14-017), and the Spanish Ministry of Economy and Competitiveness (grant ECO-2014-52519-R and TIN2009-12132).

REFERENCES

[1] A. Smith, Record shares of Americans now own smartphones, have home broadband, Pew Research Center, 2017. [Online]. Available: http://www.pewresearch.org/fact-tank/2017/01/12/evolution-of-technology/. [Accessed: 12-Jul-2017].

[2] H. Guo, H. Huang, L. Huang, and Y. E. Sun, Recognizing the operating hand and the hand-changing process for user interface adjustment on smartphones, Sensors (Switzerland), vol. 16, no. 8, pp. 129, 2016.

[3] Mobile Fact Sheet, Pew Research Center, 2017. [Online]. Available: http://www.pewinternet.org/fact-sheet/mobile/. [Accessed: 12-Jul-2017].

[4] Compuware Corporation, Mobile Apps: What consumers really need and want - A global study of consumers expectations and experiences of mobile applications, pp. 118, 2012.

[5] S. Azenkot and S. Zhai, Touch behavior with different postures on soft smartphone keyboards, Proc. 14th Int. Conf. Human-computer Interact. with Mob. devices Serv. - MobileHCI 12, p. 251, 2012.

[6] M. Löchtefeld, P. Schardt, A. Krüger, and S. Boring, Detecting Users Handedness for Ergonomic Adaptation of Mobile User Interfaces, Proc. 14th Int. Conf. Mob. Ubiquitous Multimed., pp. 245249, 2015

[7] M. Gonzalez-Rodriguez, J. Manrubia, A. Vidau, and M. Gonzalez-Gallego, Improving accessibility with user-tailored interfaces, Appl. Intell., vol. 30, no. 1, pp. 6571, 2009.

[8] A. Kos, S. Tomažič, and A. Umek, Evaluation of Smartphone Inertial Sensor Performance for Cross-Platform Mobile Applications, Sensors, vol. 16, no. 4, 2016.

[9] M. Zheng, O. Ormandjieva, and H. Fan, Designing Context Sensitive Mobile User Interface, in Proceedings of the International Conference on Software Engineering Research and Practice (SERP), 2015, p. 152.

[10] M. Goel, J. O. Wobbrock, and S. N. Patel, GripSense, Proc. 25th Annu. ACM Symp. User interface Softw. Technol. - UIST 12, p. 545, 2012.

[11] K. Seipp and K. Devlin, One-Touch Pose Detection on Touchscreen Smartphones, Proc. 2015 Int. Conf. Interact. Tabletops Surfaces - ITS 15, pp. 5154, 2015.

[12] A. Buriro, Z. Akhtar, B. Crispo, and F. Del Frari, Age, gender and operating-hand estimation on smart mobile devices, in Lecture Notes in Informatics (LNI), Proceedings - Series of the Gesellschaft fur Informatik (GI), 2016, vol. P-260, pp. 15.

[13] A. K. Anne Kaikkonen, Aki Kekäläinen, Mihael Cankar, Titti Kallio, Usability testing of mobile applications: A comparison between laboratory and field testing, J. Usability Stud., vol. 1, no. 1, pp. 416, 2005.

[14] C. Hardyck and L. F. Petrinovich, Left-handedness., Psychol. Bull., vol. 84, no. 3, pp. 385404, 1977.

[15] L. Breiman, Random forests, Mach. Learn., vol. 45, no. 1, pp. 532, 2001.

[16] I. Rish, An empirical study of the naive Bayes classifier, Int. Jt. Conf. Artif. Intell., no. January 2001, pp. 4146, 2001.

[17] T. R. Patil, Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification, Int. J. Comput. Sci. Appl. ISSN 0974-1011, vol. 6, no. 2, pp. 256261, 2013.

[18] T. M. Cover and P. E. Hart, Nearest Neighbor Pattern Classification, IEEE Trans. Inf. Theory, vol. 13, no. 1, pp. 2127, Jan. 1967.

[19] J. C. Platt, Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines, 1998.

[20] Y. Freund and R. Schapire, A desicion-theoretic generalization of on-line learning and an application to boosting, Comput. Learn. theory, vol. 55, pp. 119139, 1996.

[21] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, Assessing the accuracy of prediction algorithms for classification: an overview, Bioinformatics, vol. 16, no. 5, pp. 412424, 2000.

[22] R. Kohavi, A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection, in Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2, 1995, pp. 11371143.

[23] Worldbank, Population, female (% of total) — Data, WorldBank, 2017. [Online]. Available: http://data.worldbank.org/indicator/SP.POP.TOTL.FE.ZS?end=2017&start=1960&view=chart. [Accessed: 19-Jul-2017].