Smartphone Screen Gesture Recognition Using Smartphone Sensors

Priyanka Bhatele
School of Computer Engineering and Technology
Dr. Vishwanath Karad MIT World Peace University, PUNE
Pune, Maharastra, India
0009-0004-7637-1104

Mangesh Bedekar
School of Computer Engineering and Technology
Dr. Vishwanath Karad MIT World Peace University, PUNE
Pune, Maharastra, India
mangesh.bedekar@mitpune.edu.in

Abstract*— In the recent times, smartphone usage has become increasingly popular for learning. User's exhibit multiple gesture interactions with smartphones, while reading, which can provide valuable implicit feedback about the content consumed. Smartphones have many embedded sensors which capture plethora of user interaction data. The on-device Gyroscope and Accelerometer can be enabled to capture the variations done due to gesture interactions like scrolling, pinch to zoom, tap, orientation change and screen capture. This research work is based on training machine learning classifier models with smartphone sensors' readings to identify the users screen gesture interactions. Data for the classifier is collected by from 44 users in total using an android application. Aggregated time domain feature extraction has been computed on the preprocessed data. Four groups of data have been used to train the models. Extensive experiments are done to test the success of proposed system using Random Forests, Support Vector Machine (SVM), Extreme Gradient Boost (XGB), ADA boost, Naïve Bayes (NB) and K-Nearest Neighbour (KNN). Detailed analysis of the success rate and accuracy calculation have been performed. Best identification accuracy of 97.58% is achieved by Random Forest Classifier followed by Extreme Gradient Boost and K-Nearest Neighbour with accuracy 95.97% and 93.55% respectively.

Keywords*— Gesture recognition, Smartphone sensors, Mobile sensing, Screen gestures, Online learning, Implicit feedback.

I. Introduction

Currently there are around 400 million mobile gamers in India. This number is estimated to raise to 650 million by 2025. Embedded sensors in Smartphones and Tablets are the major contributors to support gaming [4]. Gyroscope, Accelerometer, Proximity Sensor, Camera and Electronic Compass are supporting not only gaming but many features in a smartphone that makes it a smart device. Despite making life easy for the users, embedded sensors also have a wide area of applications like Bio-Feedback, Implicit Feedback and Authentication. Gyroscope and accelerometer have the capability to capture abundant data. This data can be further analytically studied to understand the health of the (patients) user, the intent and authentication of the user.

Web Analytics is the domain that analyses the behavior of the web users. There are many factors like dwelling time, page views and page clicks that are the prime contributors for such analysis [5]. Activity inputs given through peripherals like mouse and keyboards has been another vital means for recent researches. Left to right movement of mouse is been hypothetically proved to be the pointer assisted reading and a positive indicator or Key Performance Indicator (KPI). Another significant indicator of performance is copy to clipboard activity. Data frequently copied can also has been studied to be positive for phrases, codes and paragraphs. On the contrary it has been demonstrated a negative indicator for words [2, 3, 8].

Smartphone is popular amongst the youth for reading (consuming textual content). It's one of the foremost tools for pedagogical learning [21]. It acts as a common tool for referring study materials, capturing images of lecture slides and notes [20,28]. Logging academic work portals is another common usage of smartphones among students [22]. Young scholars create notes in the form of screen captures of online content [17]. Reading online is one of the vital and foremost usage of smartphones these days. Millions of scholars are using smartphones as their learning tool. Smartphones are common among all the age groups ranging from 05yrs –to– 45yrs.

Most researchers have captured feedback using surveys, as an explicit method to understand the quality of consumed content. Implicit methods of determining the quality of the content still remains unexplored [19,20,21,22,23,24,25].

Implicit feedback methods for online reading are relatively un-explored for users using smartphones and tablets. When a user intends to read online, using the smartphone, the user will make some screen handling gestures [28] on the mobile screen. This research works proposes to capture, identify and categorize these screen activities and mobile gestures interactions using machine learning classifier algorithms. Smartphone sensor readings while doing such mobile gesture interactions can be used to train various training models. Thorough Testing and Validation of these models can be done to generate accurate results.

II. Problem Statement

When a user is reading using smartphone, the user tends to perform some screen activities like scrolling, pinch to zoom, tap, etc. Change in orientation (ideally portrait mode to landscape mode) is done for adjusting the viewport of the mobiles screen for better readability. This gesture like orientation changes from portrait to landscape can be captured. Screen capture is another activity that is common among young scholars. Smartphone embedded sensors, gyroscope and accelerometer are continuously active and collect readings without any explicit involvement from the user. This gesture data collected can be captured and analyzed for recognition of such reading activities. Training and testing on this data can be used to develop heuristics. Smartphone sensors can detection acceleration and rotational movement along the three axes using the accelerometer and gyroscope. Training of various supervised classifiers to identify these (reading) activities on smartphones can be done.

This research works is looking to close the gap of implicit feedback methods for online content read using smartphones. It contributes in following ways:

A. Dataset creation of 44 users performing mobile gesture interactions like Scrolling, Tap, Pinch to Zoom, Orientation Change and Screen Capture.

B. Design and implementation of system to identify these gesture interactions using machine learning classifier algorithms.

C. Preparation of evaluation metrics to assess the expected results of the system.

D. Perform evaluation and analysis of the performance of the system proposed.

III. Existing Literature

A. Smartphone as a Pedagogical tool

Smartphone is a common pedagogical tool for the age group range of 10-35years. [19,21] Children up to 5 years of age has been involved too in the smartphone usage for entertainment purpose. Research studies have also been done on the users of age above 35years and found to be a useful tool for e-learning.

Ubiquitous learning in one of the foremost usage of smartphones. Smartphones, Tablets and their availability in affordable price range has aided their widespread use in learning [18,20]. Smartphone embedded sensors enables functions of triggering applications using user defined gestures. Quick access to application is the proof of such functions [17]

B. *Implicit Feedback Methods in Desktop Systems*

1) Dwell time and Scrolling: Mark Claypool et.al. had studied several activities like mouse activity, keyboard activity, dwell time and scrolling activity in Kruksal-Wallis test. Keyboard, Scrolling and dwell time correlated well in degree of independence as an implicit indicator, whereas Mouse activity did not correlate well [5].

2) Copy to Clipboard: Few lines of code are able to capture the data copied in the clipboard. Clipboard data gives the insight of the user intentions in copying text. Category of data copied are copy of words, phrases, translated texts (text in any other language, other than English), sentences, code fragments [3]. Diverse category can be used to draw diverse inferences. Heat maps of the copied data identified some complex words [8]. These copy to clipboard operations can help identifying the sub-page metrics and acts as an important component for key performance indicators [24]. Complex words can be simplified or a metadata attached could be the action for better understanding of readers [25]. On the contrary frequently copied sentences can be a positive key performance indicator of a valid content and can be further utilized for functionality like - automatic text summarization [2], recommendations, etc. This data can be used to understand the users interest and also for Search engine optimization [3,24].

3) Mouse Cursor Movement: Statistical study of Horizontal mouse movement done by Kirsh et.al., validates left to right cursor movement to be the pointer assisted reading. Left to right movement is analogized as eye gaze-based reading. Right to left cursor movement is done for moving to the next line or webpage change. [26] focused on the parameters horizontal distance, horizontal direction (left to right or right to left), time frame and vertical range covered. Horizontal distance of the movement in left to right direction is approximately found to be equal to the distance of the line. Left to right movement is a quicker action as compared to right to left. The vertical range of each mouse pointer movement was at par with the distance between two lines. Frequent mouse cursor movements can act as an implicit method of user interests in reading the content [23].

C. Smartphone Sensors and its Applications

Advancement in terms of computing abilities and functionalities in smartphones is likely due to the sensors embedded in it. Most smartphones of today are equipped with sensors such as are proximity sensor, gyroscope, GPS, accelerometer, microphones, camera, ambient light sensor, and digital compass. Accelerometer, gyroscope and proximity sensors are the only sensors that do not require explicit permission from the smartphone users. Such sensors are capable of capturing data silently, implicitly. This collected data can be used in various applications using machine learning classifiers [1,4,9,16,17,22].

Recognition of physical human activities like standing, sitting, walking, running etc. is termed as biofeedback. Biofeedback is main source of information for e-health monitoring. [1] Comprehensive information about users can be captured and inferred using the smartphone sensors accelerometer and gyroscope, termed as context recognition [22]. Some other applications that can be extended and implemented using the smartphone sensor data and machine learning classifier are implicit authentication and understanding of user intent [7,17].

Most of the classifiers have been trained for the biofeedback and context recognition [1,4,22] Adequate accuracy results has been achieved by support vector machine classifier for implicit authentication [4]. Table 1 shows that Gesture Recognition remains an unexplored area of research with less accurate results [4,9,16,17].

Most of the classifiers have been trained for the biofeedback and context recognition [1,4,22] Adequate accuracy results has been achieved by support vector machine classifier for implicit authentication [4]. Gesture recognition remains an unexplored area of research with less accurate results [4,9,16,17].

TABLE I. SUMMARY OF ML CLASSIFIER ALGORITHMS-PERFOMANCE ACCURACY MATRIX USING SMARTPHONE SENSOR DATA

Multi-layer Perceptron	Biofeedback	**93%**
	Context Recognition	**98.10%**
Decision Tree	Biofeedback	**96.82%**
K-Nearest Neighbour	Biofeedback	**93.30%**
	Context Recognition	**98.77%**
Bayes Net	Biofeedback	**97.38%**
	Gesture Recognition	**64.38%**
Support Vector Machine	Biofeedback	**99.18%**
	Gesture Recognition	**99%**
	Implicit Authentication	**74.78%**
Ensemble Classification	Biofeedback	**90%**
Random Forest	Context Recognition	**98.67%**
	Gesture Recognition	**74.97%**
Neural Network	Context Recognition	**94%**

D. Data Collectors

Methods on which the entire implicit feedback model relies are copy to clipboard, mouse activity and scroll activity. From these, Clipboard data and mouse activity can be captured using a few lines of JavaScript code included in the browsers [31,32]. Research reported based on smartphone sensor data, requires data collector method with respect to the context the data is required. Biofeedback requires reading from accelerometer and gyroscope for the locomotive activities (walking/running). Sensor Monitor (Pro) has been used for data collection by Umek et. al. The application streams sensor readings from smartphone to another desktop system. The overabundance of data captured can be further used to make inferences on the user behaviors, implicit authentication, user intent and context recognition and biofeedback [1,4,6,7,9,16].

In the lieu of proposed applications, the development of an implicit data capture method on the smartphone, which can silently capture sensor readings while the user is reading/scrolling through text, is required. A PDF file reading application is conceptualized, designed and implemented for this purpose.

IV. PROPOSED METHODOLOGY

The primary objective of the proposed system is to fill the gap of implicit feedback methods in the smartphone usage for online reading. If a user reads online content with interest, then the user tends to perform one or more of the activities as illustrated in Fig. 1[28].

Screen in smartphones is termed as viewport. It is the part of the complete referred content that is currently visible on the smartphone screen. The adjustment of the data content in the viewport can be done using various screen interactions like scrolling, tapping and pinch to zoom. Whenever when a user is reading content, the user will scroll the content to adjust to the viewport. Orientation is adjusted from Portrait to Landscape or vice versa, to see the content in a better readable mode. Tap and Pinch to zoom is often done to read the content with required clarity, in case of poor visibility. Screen Capture, the activity done in case the content is to be further referred by the user. All these gestures are the indicators of user reading activity and the positive indicators of user interests in the content [28].

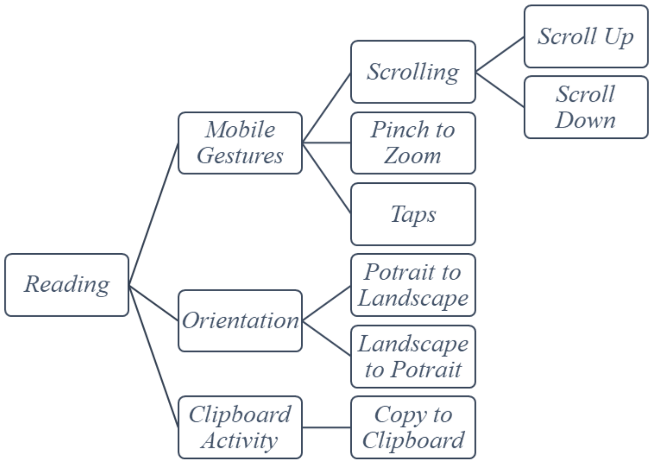

Fig. 1. User Gesture Interactions while Reading using Smartphones [28]

Smartphone Sensors generate sensor data readings without any explicit permission from the user. Readings of the screen activity usually follows a pattern This paper proposes a model to implement machine learning classifiers to identify these activities using smartphone sensors readings as the features. The model is trained on the features extracted by captured sensor data readings and trained to recognize the gesture interaction. Fig. 2 shows the implementation process of the proposed system.

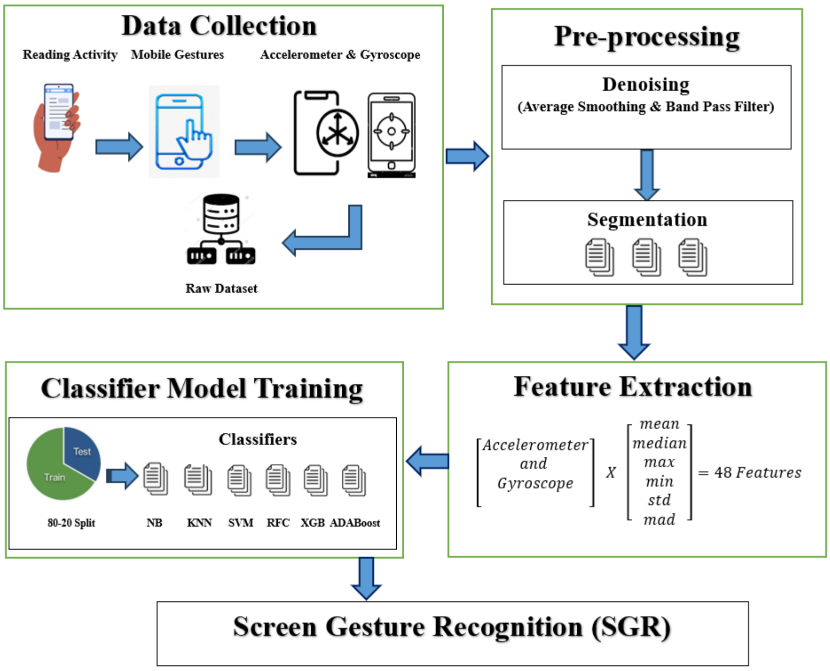

Fig. 2. Implementation Process of the Proposed System

V. IMPLEMENTATION METHODOLOGY

A. Smartphone Embedded Sensors

Smartphone have various sensors in them. Some of these sensors do not need explicit permissions from the user. Calibrating these sensors with smartphone application is capable of generating abundance of data. Android based smartphones comprise of two primary sensors to sense motion and rotation.

1) Accelerometer (Motion Sensing)
2) Gyroscope (Rotation Sensing)

Both these sensors provide measurements for x-y-z axis in a 3-D coordinate system. It generates data readings consisting of acceleration and angular velocity respectively in each direction.

$$\text{Accelerometer: } a = \{a_x, a_y, a_z\} \quad (1)$$
$$\text{Gyroscope: } g = \{g_x, g_y, g_z\} \quad (2)$$

When the smartphone is kept in idle position on a flat surface, the generated readings ideally should be A= {0, 0, ±g}. However, observed readings for accelerometer, keeping the smartphone in an idle state on a flat surface, is as shown below.

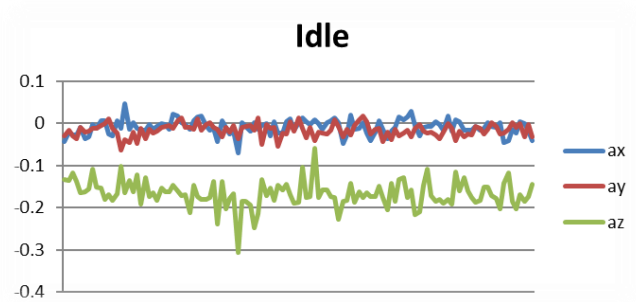

Fig. 3. Sample Accelerometer Reading when Smartphone is kept in Idle state [28]

Similar readings can be observed, in gyroscope readings while the smartphone is in an idle position. The readings are not constant, rather they show a small variation within a range. These sensors are highly sensitive, often leading to the non-zero data generation. In order to overcome the sensitivity issue, most related studies have included the fourth feature, magnitude along with x, y, and z axis [27]. The magnitude feature is calculated as below for both accelerometer and gyroscope. [7, 27]

$$\text{Accelerometer } a_{mag} = a_x^{\,2} + a_y^{\,2} + a_z^{\,2} \quad (3)$$
$$\text{Gyroscope } g_{mag} = g_x^{\,2} + g_y^{\,2} + g_z^{\,2} \quad (4)$$

Magnitude of the sensor is not affected with the orientation sensitivity. With the inclusion of the fourth feature, the final reading calibrated for both sensors are:

$$\text{Accelerometer} = \{a_x, a_y, a_z, a_{mag}\} \quad (5)$$
$$\text{Gyroscope} = \{g_x, g_y, g_z, g_{mag}\} \quad (6)$$

B. Raw Data Collection

Gesture recognition datasets also does not specifically cover gesture type while reading a document online. Hence dataset creation is the fundamental requirement to proceed further for this research work. To collect data from smartphones sensors (gyroscope and accelerometer), a PDF file reader application named *Books* has been developed using Flutter and Firebase [10,11,12,13,14,15]. The application enables reading PDF documents on smartphones and tablets. The software application is capable of capturing readings of Gyroscope and Accelerometer sensors embedded in smartphones and tablets, in the background, without any user involvement, implicitly. It captures the x y and z axes values along with the date and timestamp. In order to provide better accuracy, the magnitude feature is also calculated. Sensor data capturing rate is 1 Hz (sensor reading data is captured every second). Data logged is saved in a text file. The smartphone device used for dataset collection was OnePlus Nord2 5G. Similar instrumentations can be done on any smartphone device for capturing reading activity gestures.

47 subjects were selected for data collection of user gesture activity while reading using *Books* application [28]. The mean age of all the subjects was 22.95years. Figure 4 shows the age group wise count. The selection of high count of subjects in the age group of 15-40 years is on the basis of the existing literature of smartphone usage [1,4,6,7,9,16]. Gestures observed for the age group 5-15 years was found to be inconsistent (weird and haphazard).

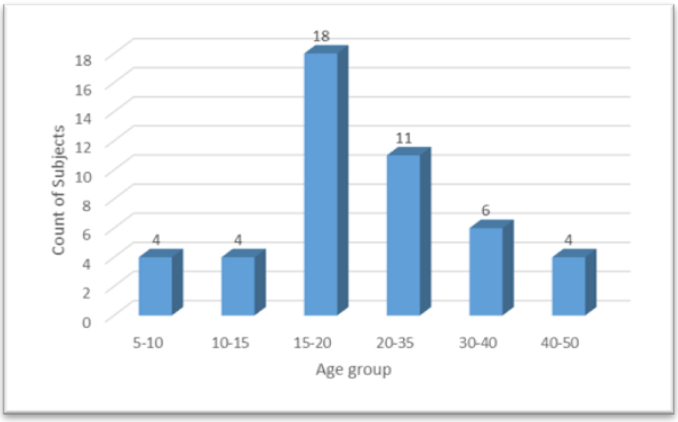

Figure 4: Age Group wise count of Subjects involved for Data Collection

Reading activity involves specific mobile gestures due to the small size screen of smartphones. Screen gesture is a pattern of touch events over the screen starting with finger down to ending up with finger up. Accelerometer and gyroscope readings were captured while performing below activities in smartphone:

1) *Screen Activity* - Screen activity clubs the mobile gestures done on the screen of the smartphone while reading. It involves:

a) *Pinch to Zoom* - Pinch-to-zoom refers to the series of touch interaction that zooms in or out the viewport to display the content on the screen. To use pinch to zoom, two fingers are moved apart to zoom in, or close-by to zoom out.

b) *Scroll Activity*- It is sliding text, images or video across a mobile screen or display, vertically or horizontally.

c) *Tap* - Tap is when you touch on the same spot for a longer time, e.g., to select an icon.

2) *Orientation* - Orientation is the horizontal or vertical positioning of the viewport for better view. For example, Portrait and Landscape are two common orientations in smartphones.

3) *Screen Capture* - Screen capture is taking a screenshot of the viewport to save it for further use. Mostly in android phones it is captured by pressing the power and volume down buttons at the same time. Some smartphones can recognize three-finger scroll-down gesture for the same functionality.

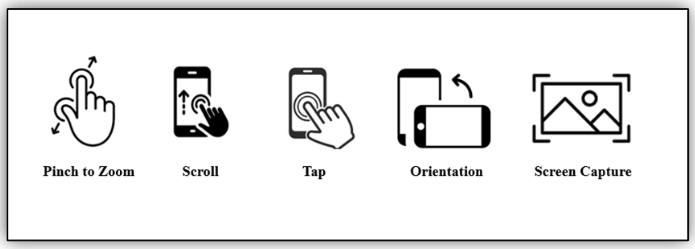

Fig. 5. Mobile Gesture Interactions used for Data Collection

The data collection activity was scheduled separately for each subject. Prior to the event, each participant was given a common roadmap as shown in figure 6.

In total, 3092 data points were generated and captured. The data was labeled as 'Screen Activity', 'Orientation' and 'Screen Capture'. Below table displays the gesture wise count plot of the complete data captured. The figure shows that the dataset is nearly balanced with almost equal values captured for the labeled gestures.

This complete activity is of 8-10 minutes and for the purpose of a research work. A pdf document will be opened using the *Book* application. Following mobile gestures and screen activities are to be performed as instructed.

1. **Scrolling** – At every 4 second you must scroll up/down the pdf document. Continue the activity for 40 seconds.
2. **Pinch to Zoom-** There are various pictures and tables given in small fonts. In order to view or read them you must *pinch to zoom in* every 4 sec for 1 minutes.
3. **Tap-** Tapping all over the document is to be done, every 4 sec for 1 minute.
4. **Orientation-** Change of orientation from "*portrait to landscape*" and "*landscape to portrait*", every 4 sec for 2 minutes.
5. **Screen Capture**: Take Screen shot every 4th second for 3 Minutes

Thanks a lot for your cooperation. Your contribution will be acknowledged.

Fig. 6. Road Map

C. Preprocessing: De-noising and Segmentation

Data collection process has been done with a smartphone device used regularly, in order to collect data intrusively. This research work could be extended to real world only if the initial research is done in an uncontrolled environment. Data collected from the smartphone sensors are prone to be noised due to sudden spikes that is caused due to messages or calls received during the data collection process. De-noising method used in references includes average smoothing [40]. Each raw data is been replaced by the average of the next two readings. In references [feature extraction] band-pass filtration been done to eliminate the gravity factor from the accelerometer sensor value. In this research raw data has been preprocessed with the smoothing and the band-pass filter steps.

Data segmentation is another preprocessing method done in references [37,38,39,40,41,42]. It smoothens data into segment samples for further feature extraction and training. Fixed and variable sized window selection has been done similar activity recognition in existing literature. Segmentation with overlapping outperforms compared without overlapping in most of the similar researches [42]. Fixed-sized window with overlapping is done in most of the activity recognition systems [40]. Segmentation methods used in this research work are window size-10 with 50% overlapping.

D. Feature Extraction and Selection

Mobile gesture recognition has limited accuracy with smartphone sensors as reported [4,9,16,17]. Accuracy calculation depends on the features used for classification. Appropriate feature selection in smartphone sensor data may lead to better accuracy of the propose recognition system. Preprocessed data from accelerometer and gyroscope is been used to extract features for model training. Features from time domain are suitable for this research as it less complexes in terms of filtering and transformations [41]. Time domain aggregated features selected for this work are: *mean, median, minimum, maximum, standard deviation and mean absolute deviation*. These aggregate functions are applied over the preprocessed data from both the smartphone sensors. Below matrices summarizes the 48 features generated.

Column major augmented matrix Gy and Acc represents the raw data collected by the two smartphone sensors used accelerometer and gyroscope, respectively,

$$Gy = \begin{array}{c} \begin{array}{ccccc} g_x & g_y & g_z & g_{mag} & \text{actions} \end{array} \\ \left[\begin{array}{cccc|c} G_{g_x 0} & G_{g_y 0} & G_{g_z 0} & G_{g_{mag} 0} & Orientation \\ G_{g_x 1} & G_{g_y 1} & G_{g_z 1} & G_{g_{mag} 1} & Orientation \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ G_{g_x 1142} & G_{g_y 1142} & G_{g_z 1142} & G_{g_{mag} 1142} & Screen\_Activity \\ G_{g_x 1143} & G_{g_y 1143} & G_{g_z 1143} & G_{g_{mag} 1143} & Screen\_Activity \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ G_{g_x 3090} & G_{g_y 3090} & G_{g_z 3090} & G_{g_{mag} 3090} & Screen\_Capture \\ G_{g_x 3091} & G_{g_y 3091} & G_{g_z 3091} & G_{g_{mag} 3091} & Screen\_Capture \end{array}\right] \end{array}$$

$$Acc = \begin{array}{c} \begin{array}{ccccc} a_x & a_y & a_z & a_{mag} & \text{actions} \end{array} \\ \left[\begin{array}{cccc|c} A_{a_x 0} & A_{a_y 0} & A_{a_z 0} & A_{a_{mag} 0} & Orientation \\ A_{a_x 1} & A_{a_y 1} & A_{a_z 1} & A_{a_{mag} 1} & Orientation \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{a_x 1142} & A_{a_y 1142} & A_{a_z 1142} & A_{a_{mag} 1142} & Screen\_Activity \\ A_{a_x 1143} & A_{a_y 1143} & A_{a_z 1143} & A_{a_{mag} 1143} & Screen\_Activity \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{a_x 3090} & A_{a_y 3090} & A_{a_z 3090} & A_{a_{mag} 3090} & Screen\_Capture \\ A_{a_x 3091} & A_{a_y 3091} & A_{a_z 3091} & A_{a_{mag} 3091} & Screen\_Capture \end{array}\right] \end{array}$$

The rows in the above matrices are utilized in segments (window size 10) and feature extraction process is done sequentially based on the aggregated features:

$mean$ (μ), is calculated for both the matrices A and G column wise as per the below equation computed by

$$\mu = \left\{ \frac{\sum_{i=r}^{r+10} a_i}{10} \right\}_{\substack{r=[0,3091] \\ r+=5}}$$

Equation for $median(\tilde{m})$:

$$\tilde{m} = \left\{ \left\{ \frac{\left(\frac{n}{2}\right)^{th} term + \left(\frac{n+1}{2}\right)^{th} term}{2} \right\}_{\substack{n=10\\ [r,r+10]}} \right\}_{\substack{r=[0,3091]\\ r+=5}}$$

min and max, features are extracted as below:

$$min = min\{[r, r+10]\}_{\substack{r=[0,3091]\\ r+=5}}$$

$$max = max\{[r, r+10]\}_{\substack{r=[0,3091]\\ r+=5}}$$

Standard Deviation *std* (σ) is the variation of the data from the $mean$ (μ) and is calculated as below,

$$\sigma = \left\{ \sqrt{\frac{\sum_{i=r}^{r+10}(a_i - \mu)^2}{10}} \right\}_{\substack{r=[0,3091]\\ r+=5}}$$

Mean absolute deviation mad is the average of the absolute values of deviation from central measure and is given by,

$$mad = \left\{ \frac{\sum_{i=r}^{r+10} |a_i - \mu|}{10} \right\}_{\substack{r=[0,3091]\\ r+=5}}$$

mean (μ) median ($\tilde{m}$) min max std(σ) mad actions

$$Gy_{f_extract} = \left[\begin{array}{cccccccccccccccccc|c}
G_{g_x_\mu0} & G_{g_y_\mu0} & \cdots & G_{g_x_\tilde{m}0} & G_{g_y_\tilde{m}0} & \cdots & G_{g_x_min0} & G_{g_y_min0} & \cdots & G_{g_x_max0} & G_{g_y_max0} & \cdots & G_{g_x_\sigma0} & G_{g_y_\sigma0} & \cdots & \cdots & \cdots & G_{g_{mag}_mad0} & Orientation \\
G_{g_x_\mu1} & G_{g_y_\mu1} & \cdots & G_{g_x_\tilde{m}1} & G_{g_y_\tilde{m}1} & \cdots & G_{g_x_min1} & G_{g_y_min1} & \cdots & G_{g_x_max1} & G_{g_y_max1} & \cdots & G_{g_x_\sigma1} & G_{g_y_\sigma1} & \cdots & \cdots & \cdots & G_{g_{mag}_mad1} & Orientation \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
G_{g_x_\mu228} & G_{g_y_\mu228} & \cdots & G_{g_x_\tilde{m}228} & G_{g_y_\tilde{m}228} & \cdots & G_{g_x_min228} & G_{g_y_min228} & \cdots & G_{g_x_max228} & G_{g_y_max228} & \cdots & G_{g_x_\sigma228} & G_{g_y_\sigma228} & \cdots & \cdots & \cdots & G_{g_{mag}_mad228} & Screen_Activity \\
G_{g_x_\mu229} & G_{g_y_\mu229} & \cdots & G_{g_x_\tilde{m}229} & G_{g_y_\tilde{m}229} & \cdots & G_{g_x_min229} & G_{g_y_min229} & \cdots & G_{g_x_max229} & G_{g_y_max229} & \cdots & G_{g_x_\sigma229} & G_{g_y_\sigma229} & \cdots & \cdots & \cdots & G_{g_{mag}_mad229} & Screen_Activity \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
G_{g_x_\mu615} & G_{g_y_\mu615} & \cdots & G_{g_x_\tilde{m}615} & G_{g_y_\tilde{m}615} & \cdots & G_{g_x_min615} & G_{g_y_min615} & \cdots & G_{g_x_max615} & G_{g_y_max615} & \cdots & G_{g_x_\sigma15} & G_{g_y_\sigma615} & \cdots & \cdots & \cdots & G_{g_{mag}_mad615} & Screen_Capture \\
G_{g_x_\mu616} & G_{g_y_\mu616} & \cdots & G_{g_{x\tilde{m}}\mu616} & G_{g_y_\tilde{m}616} & \cdots & G_{g_x_min616} & G_{g_y_min616} & \cdots & G_{g_x_max616} & G_{g_y_max616} & \cdots & G_{g_x_\sigma616} & G_{g_y_\sigma616} & \cdots & \cdots & \cdots & G_{g_{mag}_mad616} & Screen_Capture
\end{array}\right]$$

mean (μ) median ($\tilde{m}$) min max std(σ) mad actions

$$Acc_{f_extract} = \left[\begin{array}{cccccccccccccccccc|c}
A_{a_x_\mu0} & A_{a_y_\mu0} & \cdots & A_{a_x_\tilde{m}0} & A_{a_y_\tilde{m}0} & \cdots & A_{a_x_min0} & A_{a_y_min0} & \cdots & A_{a_x_max0} & A_{a_y_max0} & \cdots & A_{a_x_\sigma0} & A_{a_y_\sigma0} & \cdots & \cdots & \cdots & A_{a_{mag}_mad0} & Orientation \\
A_{a_x_\mu1} & A_{a_y_\mu1} & \cdots & A_{a_x_\tilde{m}1} & A_{a_y_\tilde{m}1} & \cdots & A_{a_x_min1} & A_{a_y_min1} & \cdots & A_{a_x_max1} & A_{a_y_max1} & \cdots & A_{a_x_\sigma1} & A_{a_y_\sigma1} & \cdots & \cdots & \cdots & A_{a_{mag}_mad1} & Orientation \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
A_{a_x_\mu228} & A_{a_y_\mu228} & \cdots & A_{a_x_\tilde{m}228} & A_{a_y_\tilde{m}228} & \cdots & A_{a_x_min228} & A_{a_y_min228} & \cdots & A_{a_x_max228} & A_{a_y_max228} & \cdots & A_{a_x_\sigma228} & A_{a_y_\sigma228} & \cdots & \cdots & \cdots & A_{a_{mag}_mad228} & Screen_Activity \\
A_{a_x_\mu229} & A_{a_y_\mu229} & \cdots & A_{a_x_\tilde{m}229} & A_{a_y_\tilde{m}229} & \cdots & A_{a_x_min229} & A_{a_y_min229} & \cdots & A_{a_x_max229} & A_{a_y_max229} & \cdots & A_{a_x_\sigma229} & A_{a_y_\sigma229} & \cdots & \cdots & \cdots & A_{a_{mag}_mad229} & Screen_Activity \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
A_{a_x_\mu615} & A_{a_y_\mu615} & \cdots & A_{a_x_\tilde{m}615} & A_{a_y_\tilde{m}615} & \cdots & A_{a_x_min615} & A_{a_y_min615} & \cdots & A_{a_x_max615} & A_{a_y_max615} & \cdots & A_{a_x_\sigma15} & A_{a_y_\sigma615} & \cdots & \cdots & \cdots & A_{a_{mag}_mad615} & Screen_Capture \\
A_{a_x_\mu616} & A_{a_y_\mu616} & \cdots & A_{a_{x\tilde{m}}\mu616} & A_{a_y_\tilde{m}616} & \cdots & A_{a_x_min616} & A_{a_y_min616} & \cdots & A_{a_x_max616} & A_{a_y_max616} & \cdots & A_{a_x_\sigma616} & A_{a_y_\sigma616} & \cdots & \cdots & \cdots & A_{a_{mag}_mad616} & Screen_Capture
\end{array}\right]$$

Combined features from both the sensors have also been used in this proposed research work. The matrix $GyAcc_{f\_extract}$ shown below shows the complete parameters included:

$$GyAcc_{f\_extract} = [\, Gy_{f\_extract} \quad Acc_{f\_extract} \,|\, actions]$$

Feature extracted from the raw dataset is non linearly separable. Linearly separable data is generally classified using data classifiers like Support Vector Machine (SVM) and Logistic regression. Data classification for nonlinear dataset is well classified by K-Nearest Neighbor, Random Forest Classifier, AdaBoost, XGBoost and CATBoost. Outlier instance identification was done using box plot visualization. Inter Quartile Range (IQR) [29] was used to handle the outlier values. Outlier instances were replaced with the respective with the lower and the upper bound limit

E. Training Classification Models

Classification models were trained with four different groups of training data to evaluate the best performance. values. Standard scaling method was applied for scaling the dataset.

Mutual Information (MI) [30,34,35] is an appropriate method for feature selection that selects features neutral and unbiased for any specific model. Similar to decision tree algorithm, Mutual Information is based on the information gain. It calculates the entropy drop and under the condition of the target value. Higher the value, higher is the correlation with the target. Features that contribute to the top 65 percentile [34] have been selected.

The details of the four groups have been given below in the Table II.

TABLE II. TRAINING GROUPS

Training Groups	Description
$raw\_dataset$	Preprocessed raw dataset
$Gy_{f\_extract}$	Features Extracted from Accelerometer sensor with 10% overlap
$Acc_{f\_extract}$	Features Extracted from Gyroscope sensor with 10% overlap
$GyAcc_{f\_extract}$	Feature extracted from both the sensors with 10% overlap

Exploratory analysis performed on the raw data and features extracted, exhibited that the captured data is nonlinear and not suitably fit for any specific model. Based on the accuracy results of previous research works [my paper and sensors], various classification models are used to train the classifiers:

***Probabilistic Models**:* Probabilistic classifier like *Naïve Bayes* is a supervised machine learning algorithm that is part of generative learning algorithms. It is based on applying Bayes Theorem with a naïve assumption that the presesnce of a features in unrelatedto presence of another.

Gaussian model has been used as it is appropraite for continous values of features following a normal distribution. Model uses the training data to calculate the probability distribution of each feature and then classifies based on the probability of the target based on the features. Estimation of the most likely class is given by:

$$\hat{y} = argmax\, P(y) \prod_{i=1}^{n} P(x_i|y)$$

Geometric Models: Geometric models like Support Vector Machine and K-Nearest Neighbors are models that use geometric concepts to classify, predict and cluster.

Support Vector Machine (SVM) is based on the idea that data points in a high-dimensional space can be represented by a lower-dimensional subspace. Kernel enables the model to convert the input values into a higher dimensional space. In Gaussian distributed dataset works best with Radial Basis Function (RBF). The RBF kernel on two samples, represented as feature vectors in some input space are given by:

$$K(x_1, x_2) = \exp\left(\frac{||x_1 - x_2||^2}{2}\right)$$

here $||x_1 - x_2||^2$ is the squared Euclidean (L2-Norm) distance between two feature vectors x_1 *and* x_2

K-nearest Neighbors (KNN) - K-nearest algorithm determines the nearest neighbors and classifies according to it. KNN is based on the assumption the function is locally constant. The algorithm has been tested on both the weighted parameters, ‘uniform’ and ‘distance’. Uniform weighs all the neighboring points equally whereas distance weighs closer neighbors heavily than further ones. Parameter grid was mapped with both the parameters. The output y is based on the average of the nearest k neigbors and is given by:

$$\hat{y}(x) = \frac{\sum_{x_i} y_i}{k}$$

***Ensemble Models**: Random Forest Classifier (RFC)* is an ensemble learning classifier model. It predicts the label based on the judgement of a group of decision trees. Each tree is trained using a subset of training data. The features used for each classifier are randomly selected subset of all the features. Final classification is the predominant outcome of the individual classifiers.

Boosting Ensemble Classifier - Weak correlations with the target classes can be converted to strong correlations using boosting ensemble methods. The classifiers, trains a unit of decision tree using a separate training sample and picked with replacement over-weighted data. Residual errors are updated and learned using the feedback from the predecessors. The proposed methodology has implemented following boosting models**:**

a) Adaptive Boosting (ADA Boost), is a sequential boosting algorithm in which the adaptive subsequent weak learners are tweaked in fovor of those instances misclassified by previous classifiers.

$$E_t = \sum_i E[F_{t-1}(x_i) + \alpha_t h(x_i)]$$

$F_{t-1}(x_i)$ is the boosted classifier built up to the previous stage.
$\alpha_t h(x_i)$ weak learner added to the final boosted classifier.
E_t is the error at t-stage classifier

b) Extreme Gradient Boosting(XG Boost) is the optimized implementation of Gradient Boosting algorithm. It optimizes the loss function by iteratively adding new trees to the ensemble. Learning rate η controls the contribution of each tree.

$$F'(x) \leftarrow F(x) + \eta * \mathrm{fk(x)}$$

where, *Fk(x)* is the prediction of k^th tree.

$$F(x) = \sum fk(x)$$

All the classifiers' models are implemented using Scikit Learn library [33,36]. Hyper parameter tuning is done for the necessary classification algorithm. Table III provides the details of it. Five-fold cross-validation [33,36] was performed on these models to reduce bias and postulate more reliable results.

TABLE III. SUMMARY OF CLASSIFIER AND HYPER PARAMETER TUNING

Name		**Classifier**
NB	Group 1	Gaussian Model
	Group 2	
	Group 3	
	Group 4	
RFC	Group 1	n_estimators': 200, 'min_samples_split': 2,'min_samples_leaf': 2, 'max_features': 'auto','max_depth': 30, 'bootstrap': True
	Group 2	n_estimators': 1400, 'min_samples_split': 2, 'min_samples_leaf': 1,'max_features': 'sqrt', 'max_depth': 80, 'bootstrap': False
	Group 3	n_estimators': 200, 'min_samples_split': 5, 'min_samples_leaf': 1, 'max_features': 'sqrt', 'max_depth': 80, 'bootstrap': False
	Group 4	n_estimators': 800, 'min_samples_split': 5, 'min_samples_leaf': 1, 'max_features': 'sqrt', 'max_depth': 60, 'bootstrap': False
SVM	Group 1	SVC(kernel="rbf")
	Group 2	
	Group 3	
	Group 4	
KNN	Group 1	weights: distance, n_neighbors: 9
	Group 2	weights: 'uniform', n_neighbors: 1
	Group 3	weights: 'uniform', n_neighbors: 1
	Group 4	weights: 'uniform', n_neighbors: 1
ADABoost	Group 1	AdaBoostClassifier(n_estimators=100, random_state=0)
	Group 2	
	Group 3	
	Group 4	
XG Boost	Group 1	n_estimators': 100, 'min_child_weight': 1, 'max_depth': 15, 'learning_rate': 0.2
	Group 2	n_estimators': 900, 'min_child_weight': 1, 'max_depth': 5, 'learning_rate': 0.05
	Group 3	n_estimators': 100, 'min_child_weight': 1, 'max_depth': 3, 'learning_rate': 0.2
	Group 4	n_estimators': 1100, 'min_child_weight': 1, 'max_depth': 3, 'learning_rate': 0.15

F. Evaluation Metrics

Performance of the machine learning models were evaluated using following measures:

1) Accuracy: The percentage of correctly classified labels. The formulae to calculate the Accuracy is as below:

$$Accuracy\ \% = \left(\frac{True\ Positives + True\ Negatives}{True\ Positives\ +\ False\ Positives\ + True\ Negatives +\ False\ Negatives}\right) * 100$$

2) F1 Score, Precision, Recall Score: An alternative metric to better analyze the performance of the model. Precision is the measure of count of correctly predicted True Positives out of all positive predictions done. Recall is the measure of count of correctly predicted True Positives out of all the actual positive values. Harmonic mean of Precision and Recall is termed as F1- score.

3) Confusion Matrix: Accuracy, measures the performance of the correctly predicted values. Combinations of predicted and true values that affect the performance of the classifiers. The table/matrix visualizes such values is Confusion Matrix. A basic confusion matrix briefs, True Positive (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN).

4) Comparison with the Existent Systems: The proposed work is based on the gestures performed by the users while reading online using smartphones. Classifier models are trained based on the features extracted from the raw dataset from the smartphone sensors. The proposed system is not been implemented yet as per the best knowledge and is inspired by the similar activity recognition systems. These existent systems study mobile gestures and use it to further hypothesize the context recognition and implicit authentication methods while performing it. Less accuracy has been the drawbacks of these systems. Comparison with the proposed work is also been evaluated in the results.

VI. Results and Discussions

The training models were trained and tested on four groups mentioned in Table II. The selection of the groups was to comprehend how suitable the smartphones sensors are to identify the screen gestures. Except for *Orientation*, other gestures are quite gentle and similar in terms of acceleration generated. Possibility of it getting confused amongst were expected to be high and therefore the selection of the best accurate process as well as the dataset for further application of the proposed system was required. After the feature extracted on some groups the features were again filtered based on the mutual information gain. The models were sequentially trained by all the group datasets.

Fig 7. Shows the performance accuracy of the classifier models for all the datasets. Accuracy of $Gy_{f\_extract}$ has been the lowest for all the classifier models. Gyroscope sensor measures is calibrated to detect orientation of the device. It senses motion including vertical and horizontal rotation. As the screen gestures involves less changes along these alignments, therefore the data collected by the gyroscope can be a supporting feature instead of the

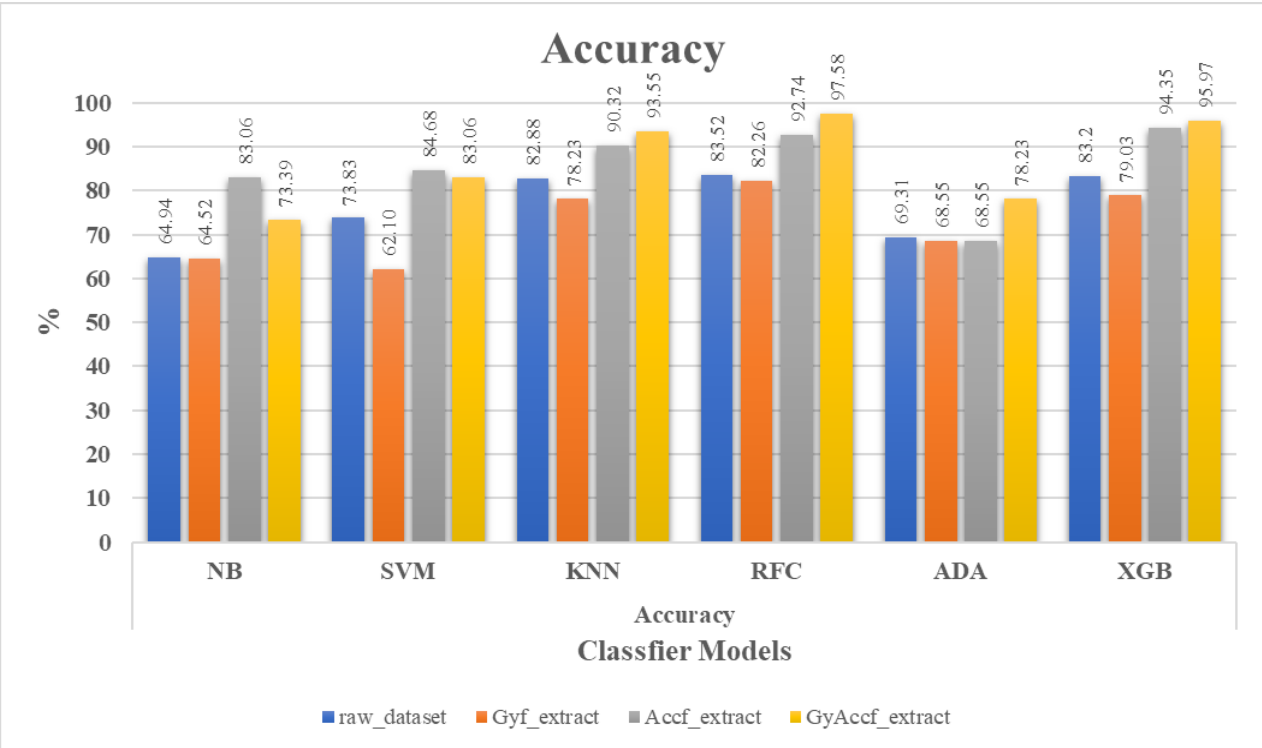

Fig. 7. Accuracy Performance of Classifier Model trained with Datasets

main training data. Overall performance of RFC is the best amongst all the classifier followed by XGB. Feature extraction in time domain is the vitality of this proposed system and it is visible through the increase in the performance accuracy from the raw pre-processed dataset (*raw_dataset*) to the feature extracted datasets ($Acc_{f\_extract}$ and $GyAcc_{f\_extract}$).

Table IV evaluates wellness of the classifier models dealing with the identification and prediction of True values. Precision and Recall evaluation parameters calculates deeper insights about performance and success rates. Uniform values of Precision and Recall justifies the accurate performance of the classifier models. Similar to the Accuracy parameter above (Fig.7) can be seen in the Precision-Recall table below.

Fig. 8 and Fig 9. shows the F score graph and the Error value graph of the classifiers with all the datasets. $GyAcc_{f\_extract}$ trains overall a highly ambiguous model that cannot be considered only as the contributor to the data for the training. Among the classifier NB and ADA Boost algorithms are unable to train well as all the parameters underperform when trained with this model. KNN algorithm performs very well the features extracted from the dataset. It can be a prominent training classifier for Screen Gesture Recognition if trained with the correct dataset. Outcomes of SVM and KNN did not resulted the similar behaviour despite being geometric models. SVM shows no better results even with appropriate dataset like KNN. Overall, the parameters could be looked upon to be improved by further feature engineering.

TABLE IV. PRECISION AND RECALL SUMMARY OF CLASSIFIER AND DIFFERENT DATASETS

Dataset \ Classifiers	NB		SVM		KNN		RFC		ADA		XGB	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Raw_Dataset	65.75	64.94	78.74	73.83	83.55	82.88	83.61	83.52	69.03	69.31	83.31	83.20
$Gy_{f\_extract}$	71.38	64.52	51.93	62.10	78.27	78.23	83.42	82.26	67.92	68.55	79.74	79.03
$Acc_{f\_extract}$	87.28	83.06	89.78	84.68	90.53	90.32	92.75	92.74	52.65	68.55	94.37	94.35
$GyAcc_{f\_extract}$	84.69	73.39	85.72	83.06	93.61	93.55	97.60	97.58	85.40	78.23	95.98	95.97

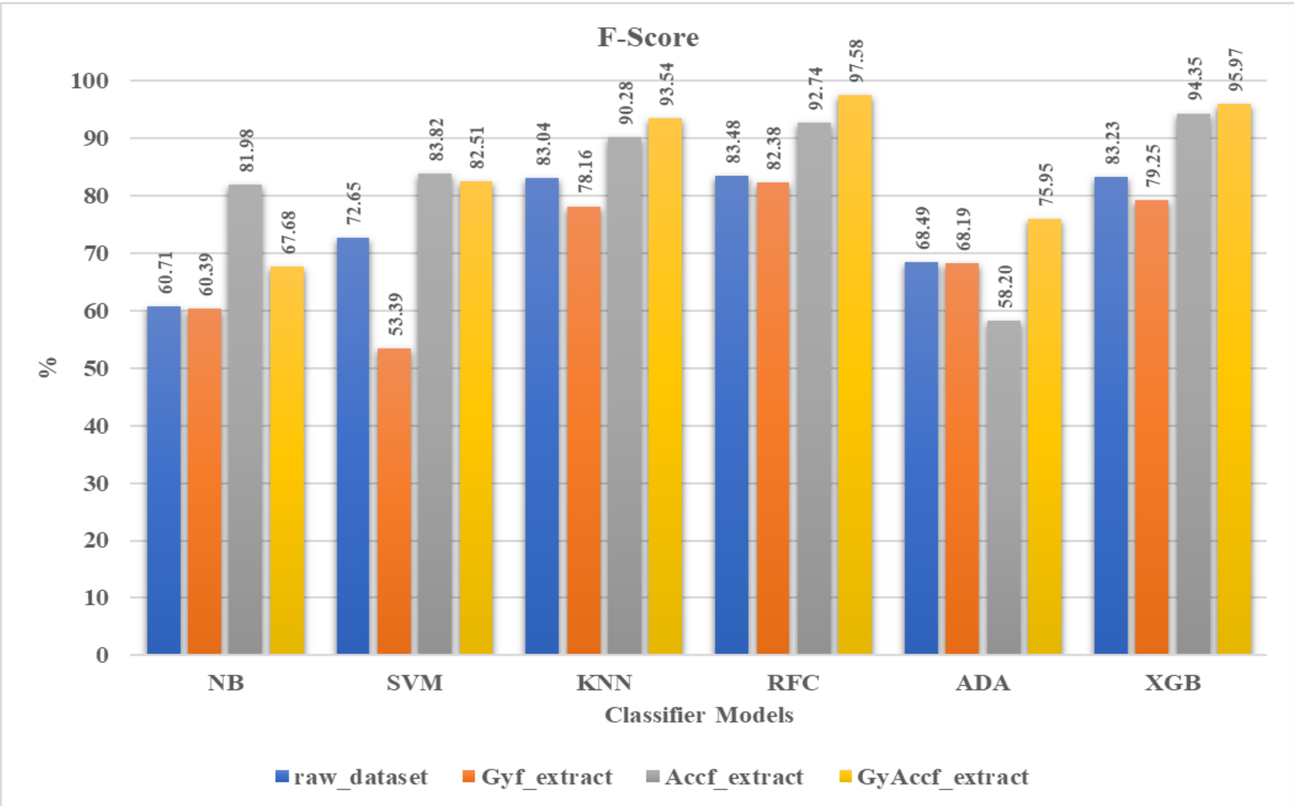

Fig. 8. F Score of Classifier Model trained with Datasets

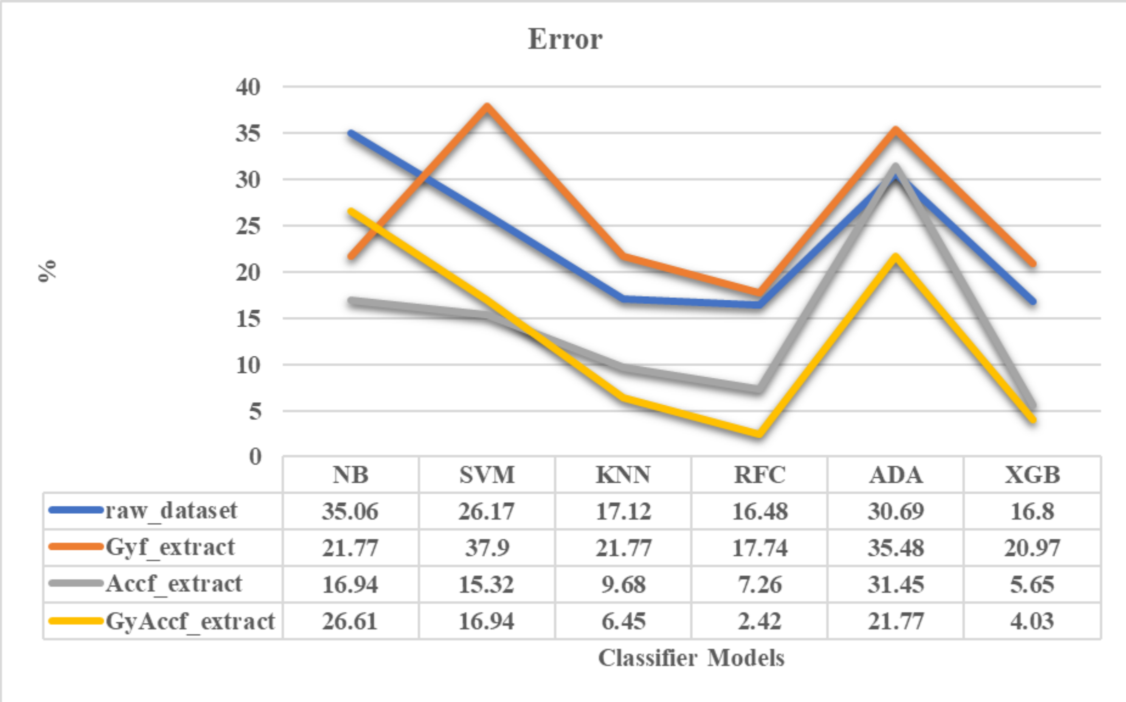

Fig. 9 Error Percentage of Classifier Model trained with Datasets

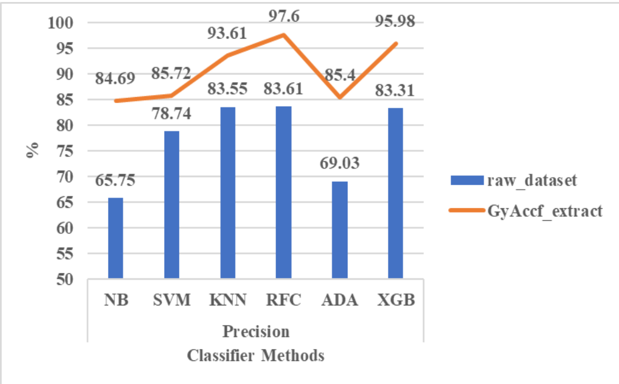

Fig. 10 Precision - *raw_dataset* vs. $GyAcc_{f\_extract}$

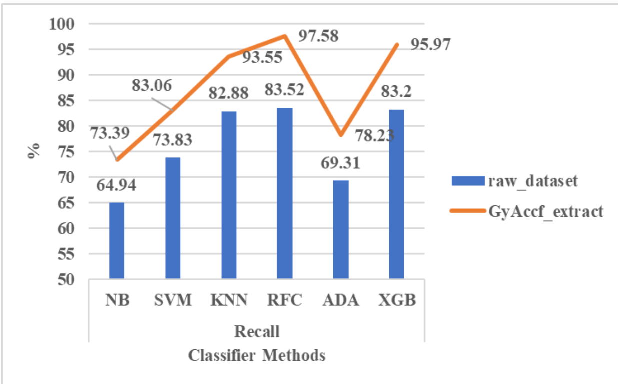

Fig. 11 Recall - *raw_dataset* vs. $GyAcc_{f\_extract}$

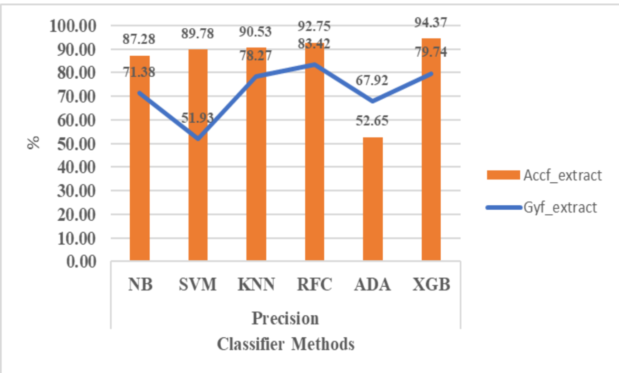

Fig. 12 Precision - $Acc_{f\_extract}$ vs. $Gy_{f\_extract}$

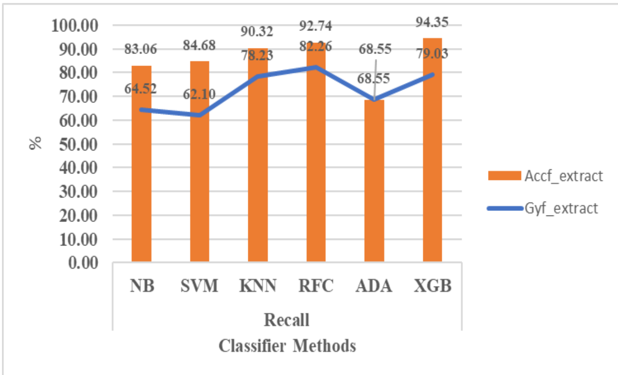

Fig. 13 Recall - $Acc_{f\_extract}$ vs. $Gy_{f\_extract}$

The proposed research focussed on four different datasets. Main objective is to improve the performance of classifier algorithms for Gesture recognition implemented in previous works. As the screen gestures considered for this study are quite similar and hence the deep insight understanding and choosing of correct dataset and the classifier to successfully identify the screen gestures while reading in smartphone. *raw_dataset* has the preprocessed data from bothe the sensors whereas $GyAcc_{f\_extract}$ contains the features extracted from it. Comparision of Precision and Recall of these two datatset has been shown in Fig. 10 and 11 respectively. Average difference in the Precision and Recall is approximately 13%. This value justifies the strength of feature extraction. Fig 12 and 13 above shows the Precision and Recall comparision between $Acc_{f\_extract}$ and $Gy_{f\_extract}$. Smartphone Screen Gestures are efficiently captured by Accelerometer as compared to Gyroscope. Yet another evaluation metrics used in this reaserch work is Confusion matrix. Confusion Matrix are has been demonstrated for the dataset $GyAcc_{f\_extract}$ as it concludes to be the best performing training dataset. Fig 14-18 shows the confusion matrix for all the classifiers trained with the datatset $GyAcc_{f\_extract}$. Confusion matrix has been plotted for each classifier model for insights about the Positive & Negative predicted values. The matrix provide a better understanding of the values. The gestures are coded as:

[*Orientation* - 0, *Screen Activity* – 1, *Screen Capture* – 2]

$$\begin{array}{cc} & \begin{array}{ccc}\mathbf{0} & \mathbf{1} & \mathbf{2}\end{array} \\ \begin{array}{c}\mathbf{0}\\ \mathbf{1}\\ \mathbf{2}\end{array} & \begin{bmatrix}46 & 0 & 0\\ 0 & 33 & 5\\ 0 & 3 & 37\end{bmatrix}\end{array}$$

Fig. 14 Confusion Matrix – KNN

$$\begin{array}{cc} & \begin{array}{ccc}\mathbf{0} & \mathbf{1} & \mathbf{2}\end{array} \\ \begin{array}{c}\mathbf{0}\\ \mathbf{1}\\ \mathbf{2}\end{array} & \begin{bmatrix}46 & 0 & 0\\ 0 & 38 & 0\\ 0 & 31 & 7\end{bmatrix}\end{array}$$

Fig. 15 Confusion Matrix – NB

$$\begin{array}{cc} & \begin{array}{ccc}\mathbf{0} & \mathbf{1} & \mathbf{2}\end{array} \\ \begin{array}{c}\mathbf{0}\\ \mathbf{1}\\ \mathbf{2}\end{array} & \begin{bmatrix}46 & 0 & 0\\ 0 & 35 & 3\\ 0 & 18 & 22\end{bmatrix}\end{array}$$

Fig. 16 Confusion Matrix - SVM

$$\begin{array}{cc} & \begin{array}{ccc}\mathbf{0} & \mathbf{1} & \mathbf{2}\end{array} \\ \begin{array}{c}\mathbf{0}\\ \mathbf{1}\\ \mathbf{2}\end{array} & \begin{bmatrix}46 & 0 & 0\\ 0 & 36 & 2\\ 0 & 1 & 39\end{bmatrix}\end{array}$$

Fig. 17 Confusion Matrix – RFC

$$\begin{array}{cc} & \begin{array}{ccc}\mathbf{0} & \mathbf{1} & \mathbf{2}\end{array} \\ \begin{array}{c}\mathbf{0}\\ \mathbf{1}\\ \mathbf{2}\end{array} & \begin{bmatrix}46 & 0 & 0\\ 0 & 38 & 0\\ 0 & 26 & 14\end{bmatrix}\end{array}$$

Fig. 18 Confusion Matrix – ADA Boost

$$\begin{array}{cc} & \begin{array}{ccc}\mathbf{0} & \mathbf{1} & \mathbf{2}\end{array} \\ \begin{array}{c}\mathbf{0}\\ \mathbf{1}\\ \mathbf{2}\end{array} & \begin{bmatrix}46 & 0 & 0\\ 0 & 35 & 3\\ 0 & 2 & 38\end{bmatrix}\end{array}$$

Fig. 19 Confusion Matrix - XGB

Fig. 14-19 clearly depicts that *Orientation* gesture is successfully identifiable by all the classifiers. The level of motion generated for an orientation change is sensitive enough to be well caliberated by the sensors and can be easily classified. The error in identification of *Screen Activity*, is not more than 13%. Underperformance of the classifiers like NB, SVM and ADA Boost is majorly because of the *Screen Capture* confused with the *Screen Activity*.

Fig 20. below shows the comparison of accuracy parameter of the existing method of Gestures with the proposed method. Existing system could classify the short term activities. The activities are dynamic and create motion horizontally and vertically. Such movements are easily classifiable using gyroscope and accelerometer as it generate notable acceleration and orientation change. It had a drawback for lesser accuracy for the gesture recognition. Proposed system achieved better accuracy due to the training and testing done with different datasets and choosing the best amongst them. Time domain feature extraction indicates to be a better approach for feature extraction for screen gesture recognition. Further process of feature selection using mutual information gain, followed by feature extraction enabled the model to identify and classify the correct gestures. The proposed system concludes to be a efficient system to classify screen gestures appropriately.

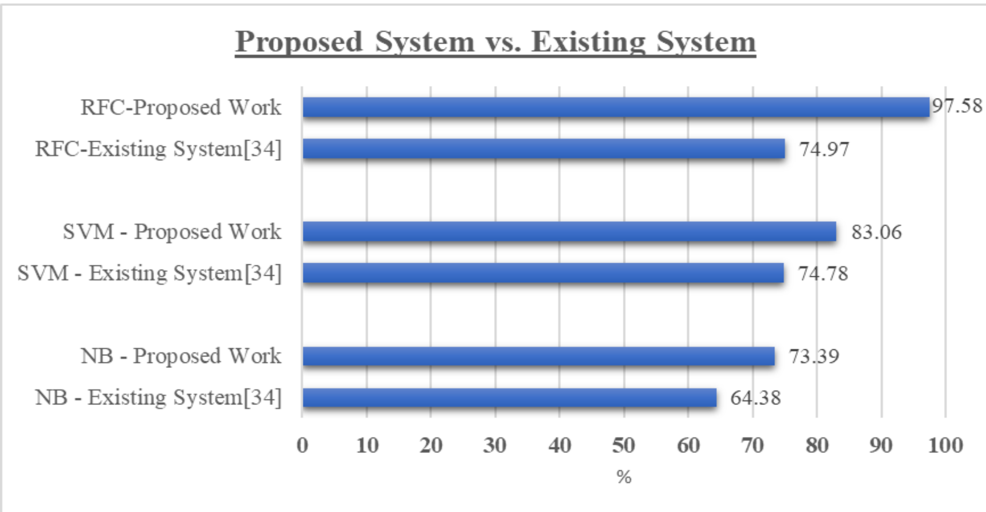

Fig. 20 Accuracy Comparision between the Proposed System (Screen Gesture Recognition System) and Existing System (Gesture Recognition)[34]

VII. Conclusion

Smartphone users performing reading activity perform mobile gestures. Smartphone embedded sensors, Accelerometer and Gyroscope capture readings while these gestures are performed. This research work is focused to identify such mobile gestures in order to create an implicit feedback model. The Mobile gestures considered in the experimentation are Screen Activity (Tap, Pinch to Zoom and Scrolling), Orientation and Screen Capture. Random Forests, Support Vector Machine (SVM), XG Boost**,** ADA boost**,** Naïve Bayes (NB) and K-Nearest Neighbour (KNN**)** models were trained on the data collected by smartphone sensors. The proposed system was trained and tested with different dataset. The procedure followed included data collection, data preprocessing, feature extraction and feature selection. Classifier models were then trained for various screen gestures. The Evaluation metrics used are Accuracy, Precision, Recall, F score values and Confusion Matrix. Proposed system was also compared with the existing system of gesture recognition. Analysis of the Accuracy and other parameters exhibits Random Forest Classifier (RFC) classifying with best accuracy results of 97.58% followed by XG Boost with 95.97% and KNN with 93.55%. Precision-Recall-F score values are observed to be in accordance with the Accuracy results. *Screen Activity* and *Screen Capture* gestures are ambiguous. Appropriate dataset to go further resulting the best results is $GyAcc_{f\_extract}$. Proposed system classifies better than the existing system and outperforms by 11% more accurracy with SVM and NB. RFC provides 24% better accuracy as compared to existing work. Proposed research contributes to identify such Screen Gestures using smartphone sensors. It does not explicitly need require permissions from the user and hence are accessible to recognize gestures.

Extension to the research work can include implementation of other classifier algorithms for training the models. Data collection can be increased for improved accuracy. Frequency domain feature extraction could also be implemented for better results. The results of this research work can be combined with it and a standalone implicit feedback system can be built.

VIII. References

1. Umek, A., Kos, A. (2016) Validation of smartphone gyroscopes for mobile biofeedback applications. Personalised and Ubiquitous Computing, 20, Pp. 657–666. https://doi.org/10.1007/s00779-016-0946-4
2. Kirsh, I. and Joy, M. (2020c). An HCI approach to extractive text summarization: Selecting key sentences based on user copy operations. In Proceedings of the 22nd HCI International Conference (HCII 2020), Communications in Computer and Information Science, Cham. Springer International Publishing. https://doi.org/10.1007/978-3-030-60700-5_53
3. Kirsh, I. (2020). What Web Users Copy to the Clipboard on a Website: A Case Study, WEBIST, In Proceedings of the 16th International Conference on Web Information Systems and Technologies (WEBIST 2020), SciTePress, INSTICC, 2020. August 2020 http://dx.doi.org/10.5220/0010113203030312
4. Liu, M. (2013). A Study of Mobile Sensing Using Smartphones. International Journal of Distributed Sensor Networks. https://doi.org/10.1155/2013/272916
5. Mark Claypool, Phong Le, Makoto Wased, and David Brown (2001). Implicit interest indicators. In Proceedings of the 6th international conference on Intelligent user interfaces (IUI '01). Association for Computing Machinery, New York, NY, USA, Pp. 33–40. https://doi.org/10.1145/359784.359836
6. Voicu R-A, Dobre C, Bajenaru L, Ciobanu R-I. (2019) Human Physical Activity Recognition Using Smartphone Sensors. Sensors. 2019; 19(3):458. https://doi.org/10.3390/s19030458
7. Ehatisham-ul-Haq M, Azam MA, Loo J, Shuang K, Islam S, Naeem U, Amin Y. (2017) Authentication of Smartphone Users Based on Activity Recognition and Mobile Sensing. Sensors; 17(9):2043. https://doi.org/10.3390/s17092043
8. Kirsh, I. and Joy, M. (2020a). A different web analytics perspective through copy to clipboard heatmaps. In Proceedings of the 20th International Conference on Web Engineering (ICWE 2020), Lecture Notes in Computer Science, Vol 12128, pp. 543–546, Cham. Springer International Publishing. http://doi.org/10.1007/978-3-030-50578-3_41
9. N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury and A. T. Campbell (2010) A survey of mobile phone sensing, in IEEE Communications Magazine, Vol. 48, No. 9, Pp. 140-150, https://doi.org/10.1109/MCOM.2010.5560598
10. Payne, R. (2019). Using Firebase with Flutter. In: Beginning App Development with Flutter. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-5181-2_12
11. Tashildar, A., Shah, N., Gala, R., Giri, T., & Chavhan, P. (2020). Application development using flutter. International Research Journal of Modernization in Engineering Technology and Science, 2582–5208. http://www.irjmets.com/uploadedfiles/paper/volume2/issue_8_august_2020/3180/1628083124.pdf
12. Bhagyashree Solegaonkar | Yogeshchandra Puranik "Fun with Flutter" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-5 | Issue-4, June 2021, pp.975-977, URL: www.ijtsrd.com/papers/ijtsrd42449.pdf
13. Khawas, Chunnu & Shah, Pritam. (2018). Application of Firebase in Android App Development-A Study. International Journal of Computer Applications. 179. Pp. 49-53. https://doi.org/10.5120/ijca2018917200.
14. Wasilewski K, Zabierowski W. (2021) A Comparison of Java, Flutter and Kotlin / Native Technologies for Sensor Data-Driven Applications. Sensors. 2021; 21(10):3324. https://doi.org/10.3390/s21103324
15. Moroney, L. (2017). The Firebase Realtime Database. In: The Definitive Guide to Firebase. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-2943-9_3.
16. Cai, Liang & Machiraju, Sridhar & Chen, Hao. (2009). Defending against sensor-sniffing attacks on mobile phones. 31-36. 10.1145/1592606.1592614 .
17. Li, Qiang & Han, Qi & Sun, Limin. (2014). UserIntent: Detection of user intent for triggering smartphone

sensing applications, Pp. 185-187. https://doi.org/10.1109/SAHCN.2014.6990351
18. Park, Y. and Chen, J.V. (2007), "Acceptance and adoption of the innovative use of smartphone", Industrial Management & Data Systems, Vol. 107 No. 9, Pp. 1349-1365. https://doi.org/10.1108/02635570710834009
19. Dong-Hee Shin, Youn-Joo Shin, Hyunseung Choo, Khisu Beom (2011) Smartphones as smart pedagogical tools: Implications for smartphones as u-learning devices, Computers in Human Behavior, Volume 27, Issue 6, Pp. 2207-2214, ISSN0747-5632, https://doi.org/10.1016/j.chb.2011.06.017
20. S. Saraswathi. (2017). Smartphone usage among students. International Education and Research Journal, Vol 3, No 6.
21. Singh, Anjali and Dinesh Kumar Srivastava. (2021) Understanding the Effect of Internet Addiction on Student Academic Engagement, IJICTE Vol.17, No.4, Pp.1-12. http://doi.org/10.4018/IJICTE.20211001.oa11
22. Abayomi Moradeyo Otebolaku, Maria Teresa Andrade (2016) User context recognition using smartphone sensors and classification models, Journal of Network and Computer Applications, Volume 66, Pp. 33-51, ISSN 1084-8045,https://doi.org/10.1016/j.jnca.2016.03.013
23. Kirsh, I. (2020). Directions and Speeds of Mouse Movements on a Website and Reading Patterns: A Web Usage Mining Case Study. Proceedings of the 10th International Conference on Web Intelligence, Mining and Semantics. June 2020, Pp 129–138. https://doi.org/10.1145/3405962.3405982
24. Kirsh, I., & Joy, M. (2020). Splitting the Web Analytics Atom: From Page Metrics and KPIs to Sub-Page Metrics and KPIs. Proceedings of the 10th International Conference on Web Intelligence, Mining and Semantics. June 2020, Pp. 33–43, https://doi.org/10.1145/3405962.3405984
25. Kirsh, I. (2020a). Automatic complex word identification using implicit feedback from user copy operations. In Proceedings of the 21st International Conference on Web Information Systems Engineering (WISE 2020), Lecture Notes in Computer Science, forthcoming, Cham. Springer International Publishing. https://doi.org/10.1007/978-3-030-62008-0_11
26. Kirsh, I., Joy, M., &Kirsh, Y. (2020). Horizontal Mouse Movements (HMMs) on Web Pages as Indicators of User Interest. (HCI International 2020), CCIS, Springer, 2020. https://doi.org/10.1007/978-3-030-60700-5_53
27. Olsen, A.F. (2016) Detecting Human Emotions Using Smartphone Accelerometer Data. Master's Thesis, Department of Informatics, University of Oslo, Oslo, Norway, 2016.
28. P. Bhatele and M. Bedekar, (2023) Survey on Smartphone Sensors and User Intent in Smartphone Usage, 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), Lonavla, India, 2023, Pp. 1-9, doi: 10.1109/I2CT57861.2023.10126192.
29. Taylor, Courtney (2020). What Is the Interquartile Range Rule? ThoughtCo, Aug. 26, 2020, https://www.thoughtco.com/what-is-the-interquartile-range-rule-3126244 .
30. Zhu, Andrew. (2021) Select Features for Machine Learning Model with Mutual Information. Towards Data Science, Jun 29, 2021, https://towardsdatascience.com/select-features-for-machine-learning-model-with-mutual-information-534fe387d5c8
31. Saniya Zahoor, Mangesh Bedekar, and Pranali K. Kosamkar (2014). User Implicit Interest Indicators learned from the Browser on the Client Side. In Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies (ICTCS '14). Association for Computing Machinery, New York, NY, USA, Article 57, 1–4. https://doi.org/10.1145/2677855.2677912
32. Bedekar, Mangesh & Zahoor, Saniya. (2014). Implicit Client Side User Profiling for Improving Relevancy of Search Results. Computer Science & Information Technology. 4. 10.5121/csit.2014.4809.
33. Alqarni, Mohammed & Chauhdary, Sajjad & Naseer Malik, Maryam & Ehatisham-ul-Haq, Muhammad & Azam, Muhammad Awais. (2020). Identifying smartphone users based on how they interact with their phones. Human-centric Computing and Information Sciences. doi: 10. 10.1186/s13673-020-0212-7.
34. N. Kwak and Chong-Ho Choi, (2002). Input feature selection by mutual information based on Parzen window, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 12, pp. 1667-1671, Dec. 2002, doi: 10.1109/TPAMI.2002.1114861.
35. Guhanesvar. (2021) Feature Selection Based on Mutual Information Gain for Classification and Regression. Medium, Jun 26, 2021, https://guhanesvar.medium.com/feature-selection-based-on-mutual-information-gain-for-classification-and-regression-d0f86ea5262a
36. Lau, D.S., Ajoodha, R. (2022). Music Genre Classification: A Comparative Study Between Deep Learning and Traditional Machine Learning Approaches. In: Yang, XS., Sherratt, S., Dey, N., Joshi, A. (eds) Proceedings of Sixth International Congress on Information and Communication Technology. Lecture Notes in Networks and Systems, vol 217. Springer, Singapore. https://doi.org/10.1007/978-981-16-2102-4_22
37. Maryam Banitalebi Dehkordi, Abolfazl Zaraki, Rossitza Setchi, Optimal Feature Set for Smartphone-based Activity Recognition,Procedia Computer Science,Volume 192,2021,Pages 3497-3506,ISSN 1877-0509,https://doi.org/10.1016/j.procs.2021.09.123.
38. Daniel Garcia-Gonzalez, Daniel Rivero, Enrique Fernandez-Blanco, Miguel R. Luaces, New machine learning approaches for real-life human activity recognition using smartphone sensor-based data, Knowledge-Based Systems, Volume 262,2023,110260, ISSN 0950 7051, https://doi.org/10.1016/j.knosys.2023.110260.
39. X. Su, H. Tong and P. Ji, "Activity recognition with smartphone sensors," in Tsinghua Science and Technology, vol. 19, no. 3, pp. 235-249, June 2014, doi: 10.1109/TST.2014.6838194.
40. Dargie, Waltenegus. (2009). Analysis of Time and Frequency Domain Features of Accelerometer

Measurements. Proceedings - International Conference on Computer Communications and Networks, ICCCN. 1 - 6. 10.1109/ICCCN.2009.5235366.

41. Hirawat, Amit & Taterh, Swapnesh & Sharma, Dr. Tarun. (2021). A dynamic window-size based segmentation technique to detect driver entry and exit from a car. Journal of King Saud University - Computer and Information Sciences. 34. 10.1016/j.jksuci.2021.08.028.
42. Baldominos, Alejandro & Cervantes, Alejandro & Sáez, Yago & Isasi, Pedro. (2019). A Comparison of Machine Learning and Deep Learning Techniques for Activity Recognition using Mobile Devices. Sensors. 19. 521. 10.3390/s19030521.