Article

Categorizing Touch-Input Locations from Touchscreen Device Interfaces via on-Board Mechano-Acoustic Transducers

Kai Ren Teo ^{1,*}, Balamurali B T ^{2,†}, Jianying Zhou ^{1,†} and Jer-Ming Chen ^{2,†}

¹ Information Systems Technology and Design (ISTD), Singapore University of Technology & Design, 8 Somapah Rd, Singapore 487372, Singapore; jianying_zhou@sutd.edu.sg

² Science, Mathematics and Technology (SMT), Singapore University of Technology & Design, 8 Somapah Rd, Singapore 487372, Singapore; balamurali_bt@sutd.edu.sg (B.B.T.); jerming_chen@sutd.edu.sg (J.-M.C.)

* Correspondence: kairen_teo@alumni.sutd.edu.sg

† These authors contributed equally to this work.

Abstract: Many mobile electronics devices, including smartphones and tablets, require the user to interact physically with the device via tapping the touchscreen. Conveniently, these compact devices are also equipped with high-precision transducers such as accelerometers and microphones, integrated mechanically and designed on-board to support a range of user functionalities. However, unintended access to these transducer signals (bypassing normal on-board data access controls) may allow sensitive user interaction information to be detected and thereby exploited. In this study, we show that acoustic features extracted from the on-board microphone signals, supported with accelerometer and gyroscope signals, may be used together with machine learning techniques to successfully determine the user's touch input location on a touchscreen: our ensemble model, namely the random forest model, predicts touch input location with up to 86% accuracy in a realistic scenario. Accordingly, we present the approach and techniques used, the performance of the model developed, and also discuss limitations and possible mitigation methods to thwart possible exploitation of such unintended signal channels.

Citation: Teo, K.R.; B T, B.; Zhou, J.Y.; Chen, J.-M. Categorizing Touch-Input Locations from Touchscreen Device Interfaces via on-Board Mechano-Acoustic Transducers. *Appl. Sci.* **2021**, *11*, 4834. <https://doi.org/10.3390/app11114834>

Received: 19 April 2021

Accepted: 18 May 2021

Published: 25 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: mobile input detection; mechano-acoustic fingerprint; machine learning; smartphone; touchscreen

1. Introduction

The availability of high-precision sensors such as cameras, accelerometers and microphones on modern mobile devices afford users a wide range of functionality such as navigation, virtual assistants and even pedometers. While on-board sensors enable rich user experiences, these sensors can be exploited by malicious applications to monitor the user in unintended ways by tracking transducer signals emanating from the device, such as electrical, sound and vibration signals, and often contain information about the device processes, operation and user interactions. These "collateral" signals have significant implications in the field of cyber-security and have been used to bypass cryptographic algorithms such as RSA (Rivest–Shamir–Adleman) [1] and exploit acoustic information to extract sensitive information such as retrieve user PIN codes [2] and passwords [3]. Touchscreens, the de facto user input interface, take up a significant portion of the device physical surface. Consequently, user interactions with these touchscreens generate non-negligible signals that can be recorded by on-board sensors.

On-board motion sensors such as the gyroscope and accelerometer found on mobile devices are particularly sensitive to force and direction changes when a user interacts with the device by tapping the screen. Analysing the gyroscope and accelerate sensor readings, off-the shelf mobile applications (apps) such as TouchLogger are able to infer user text inputs on various mobile operating systems (iOS and Android) with different device form factors and physical design. Extending this work on hardware sensors, ACCessory is

another application built to evaluate text input using a predictive model to infer character sequences from accelerometer data with supervised learning techniques [4].

Earlier studies analysed acoustic excitation to retrieve user input on non-touchscreen devices such as physical computer keyboards. It was demonstrated that keystroke inference can be performed using multiple microphones with relatively high accuracy [5] even when facilitated via Voice-over-IP (VoIP) services such as Skype [6]. Consequently, it is conceivable that such unauthorized audio recording can be used to recover sensitive user information, using inter-keystroke timing or statistical analysis to recover typed text [7] or even ten-character passwords within 20 attempts [8]. While some practitioners may dismiss the use of acoustic signals as a possible security loophole on mobile devices [9], recent publications [10–14] show that acoustic techniques such as tracking the Doppler effect, supplying an external excitation signal and Time Difference of Arrival (TDOA) that can be used to retrieve text input on physical keyboard can be adapted to compromise mobile devices as well. One such system, SonarSnoop utilises an active acoustic technique by emitting human inaudible acoustic signals and recording the echo to profile user interaction and infer touchscreen unlock patterns [14]. Comparably, passive techniques like TDOA which calculate the time difference between the reception of the signal by different transducers to infer input location can be further enriched with acoustic frequency analysis to distinguish touch input [13]. In this study, we apply machine learning techniques to predict the touchscreen input location from touchscreen device interfaces via acoustic fingerprints collected from on-board mechano-acoustic transducers.

An acoustic fingerprint is a summary of acoustic features extracted from an acoustic signal that can identify similar acoustic events [15]. Acoustic fingerprinting is often combined with statistical methods to identify similar types of sounds quickly and has seen application in a broad range of arenas from identifying pop music [16] to determining volcanic eruptions which inject ash into the tropopause [17].

Leveraging insights from various studies on acoustic signals from keystroke clicks on physical keyboards [5], we explore the use of keystroke inference and acoustic fingerprinting techniques on touchscreens. Using the mobile device's on-board microphones, we surmise that acoustic signals arising from the interactions with the touchscreen can reveal the user's touch input location and can thus be used to eavesdrop sensitive input information. Such a pathway may inadvertently allow user data input associated with screen input location to be inferred without users noticing [18]. We extract acoustic features from the on-board microphone signals, supported with accelerometer and gyroscope movement data to separate and classify user touch input location on the touchscreen. Our contributions in this study are twofold. Firstly, we generate a dataset containing 2-channel (stereo) audio recordings and movement data of user touch input location on a touchscreen surface under both controlled and realistic conditions. Secondly, we compare the performance of acoustic features and movement data in categorizing touch input location using machine learning algorithms.

To address the acoustic side channel presented in this paper one can consider mitigation techniques which can be broadly classified into three categories: prevention, jamming and shielding. Prevention techniques limit access to device sensors which can be implemented on both the hardware level with physical switches and in software with user access control policies. Jamming typically involves saturating sensor with noise or false information to mask the actual sound created by the touch input. Side-channel leakage can also be attenuated via shielding with physical means by altering or redistributing mass to guard against acoustic side channels.

Accordingly, this paper is organized as follows: Detection of different sources of touch input and the data collection approach used is described in Section 2. In Section 3, we describe experiments conducted regarding the extracted features and the classification process. Section 4 contains results produced as part of this investigation. We discuss the results of various sensors and possible mitigation measures in Section 5 and finally conclusions in Section 6.

2. Methodology

In this study, we record user touch input on a touchscreen and capture the corresponding physical emanations with on-board sensors. A customised Android application was adapted [19] to capture different input layouts and capture data from the hardware motion sensors and audio input. Acoustic features are extracted from the audio recordings and sensor data are categorised by touch input location. We apply machine learning techniques to the movement data and acoustic feature datasets to train separate models for each experiment. We evaluate the performance of each model and investigate the underlying phenomena that contributes to the accuracy of the model. To improve model accuracy, stereo microphone input is used and the relevant audio segments extracted with peak detection techniques. Several experiments were conducted to explore the accuracy and robustness of the selected features. Building upon prior work in this domain, we examine the use of movement data as a predictor for touch input in a realistic scenario. In addition to motion sensor such as accelerometers used in [9], we delve into the use of acoustic features for touch input classification under the same conditions and identify salient features contributing to touch input identification with a reduced feature set. Finally, we consider the relation of physical distance and separation efficiency by restricting the touch input location on the reduced touchscreen.

To investigate, we conducted a number of experiments using on-board sensors to capture the physical emanations of interactions with the touchscreen, labeled according to the corresponding touch input location.

The experiments were conducted on a Samsung Galaxy S7 Android mobile phone (SM-G930FD) using the entire touch-screen interface, 110×66 mm and subsequently with a portion of the screen, 43×66 mm divided equally into nine separate touch-input locations of three rows (Top, Mid, Bottom) and columns (Left, Mid, Right) as shown in Figure 1. An Android application was adapted to capture data from hardware motion sensors and record acoustic signals from the on board microphones. Sensor data corresponding to each of the nine locations are recorded in sequential sessions and categorised by touch-input location. As we expected to process multiple acoustic and sensor datasets, we opted to create an automated data processing pipeline to retrieve and conduct feature extraction on the recorded microphone signal to guarantee consistency across different recording sessions.

Figure 1. Touch input locations.

2.1. Recording Movement Data

To evaluate the feasibility of user touch input detection with movement data, we collected various hardware motion sensor signals accessible using our Android application.

The Android platform provides application with access to multiple environmental sensors available on the specific device model. For the purpose of our experiments, we extracted Linear Accelerometer $L(x)$, $L(y)$, $L(z)$, Gyroscope $G(y)$, $G(x)$, $G(z)$ and composite Rotation Vector $R(x)$, $R(y)$, $R(z)$ signals, recorded at 1 kHz sampling rate to capture changes in physical state of the device.

2.2. Recording Audio Data

During the recording session, dual on-board microphones located near the top and bottom of the device (see Figure 2) are used to record the stereo microphone input, captured at 44.1 kHz sampling rate in 16-bit pulse-code modulation (PCM) format. Interactions with the touchscreen create a mechano-acoustic response presenting as an impulse in the continuous microphone signal as shown in Figure 3. We record and label touch inputs for each location (Top-Left, Mid-Right etc.) in separate sessions to create the corresponding training and test datasets.

Figure 2. Device microphone locations. Note that the microphone placement deviates from the central axis and is position asymmetrically on the top and bottom of the device.

Figure 3 shows the signal from the top and bottom microphones are out of phase, alluding to the opposite orientation of the microphone membranes receiving the pulse which distinguishes the two signals. The difference between the amplitude of the pulse hints at the different degrees of mechanical coupling and internal gain of the two microphones. The difference in amplitude in the signals could be attributed to the proximity of one microphone to the source of the pulse compared to the other. We further note that the initial transient in the top signal contains a sharp spike absent from the bottom signal while a larger difference is observed between the first maxima and minima of the bottom signal. The bottom signal also has a longer tail with more oscillations after the pulse while top signal fall away quickly. With a single microphone signal, the pulse generated from a tap contains information about the touch input location based on how the sound waves propagate through the device, with different locations causing the phone to vibrate and respond uniquely (the phone's interior structure is not homogeneous). The fact that pulses observed in the top and bottom signals do not simply reflect each other suggests that the signal received at the two microphones are in fact unique. Furthermore, aftershocks with multiple peaks reflect the complex interactions between the finger and mechanical response of the phone which could include internal mechanical reflections. These repeated

peaks, may have unique magnitudes and time intervals associated with the location of the touch input as listened from two microphones positioned asymmetrically on the device. From these observations, we posit that the top and bottom microphones hear a different acoustic signal, distinguishable in their response from the same pulse which supports our assumption of uniqueness of the two signals. Our approach hinges on the uniqueness of the response of each microphone but it is the systematic difference between the two that allows our machine learning model to distinguish the location of the touch input.

Figure 3. Typical acoustic waveform from tapping the Mid-Left location: Top waveform—Top microphone (blue), Bottom microphone (red). Every tick on the horizontal axis is 10 milliseconds.

To identify and isolate the segments of acoustic signal that correlate with touch input location, we apply smoothing (moving average of 11 samples) before peak detection on the acoustic recording as seen in Figure 4, to ensure impulse-like signals are detected. A local peak detection algorithm is applied with a sliding window to select the local maximum element from the neighbourhood of the chosen frame. These detected peaks are compared against the entries from a local minimum detection algorithm applied on the original signal to exclude areas of elevated signal which are non-impulsive (plateaus) from our peak detection dataset. We apply an empirically determined peak window size of 5 k samples points (110 ms) and peak intensity threshold of 20% determined empirically.

Figure 4. Signal peaks detected in the acoustic signal. Every tick on the horizontal axis is 500 milliseconds.

Acoustic features commonly used in audio recognition and audio classification problems were extracted using an Open-Source Python Library for acoustic Signal Analysis `pyAudioanalysis` [20]. The acoustic signals are divided into frames of 27 ms (1200 samples, empirically determined to encompass the duration of a typical touch input impulse, cf. Figure 3) and for every frame a number of ‘short term’ features are extracted. These features include mel frequency cepstral coefficients (MFCCs), chroma vectors, zero crossing rate (ZCR), energy, energy entropy, spectral entropy, spectral flux, spectral roll-off, spectral spread, spectral centroid and chroma deviation. Altogether, 34 acoustic features (13 MFCCs, 13 Chromas, five spectral features, energy, entropy and zero crossing rate) were extracted for every frame totalling 68 features from both acoustic channels. The details regarding these features can be found in [20,21].

Two noteworthy sets of acoustic features used in training are MFCCs and chroma vectors with significant contribution towards touch input categorization. MFCCs focus on the perceptually relevant aspects of the audio spectrum commonly used speech/speaker recognition estimating energy in various region of audio spectrum over a set of overlapped non-linear mel-filter bank. Chroma vectors characterise energy distribution across cyclical frequency bins [22].

To facilitate feature extraction, we determine a time window around the detected impulse, distinguishing and excluding it from regions of silence or noise. Starting from the detected peak, an empirically determined time buffer is applied backwards to include the start of the impulse. ZCR across a sliding window is used to identify the start of the impulse: the start of an impulse is associated with a low ZCR value as shown in Figure 5, where we apply an empirically determined ZCR threshold of 0.015.

Figure 5. Acoustic signal (left) and typical plot of Zero Crossing Rate (ZCR) values (right) for the selected region in the signal. ZCR values are typically quite large in region of silence (ambient noise) and decreases when a tap occurs creating an impulse.

To ensure we capture the whole impulse, we begin the window with an offset from the start of the ZCR index. An empirically determined window size of 1200 samples (27 ms) with an offset ratio of 30% of the sample number between the initial index and ZCR start (shown in Figure 6) is used to encompass the entire impulse, ensuring important signal features are captured.

Figure 6. Signal window for left (blue) and right (orange) channels showing how the offset index is applied. (1) The peak is detected, then (2) a sample buffer is applied backwards, which (3) allows us to detect a drop in ZCR, thereby defining the start of the tapping impulse. To ensure we capture the whole impulse event, (4) we begin the window 30% from the start of the ZCR index.

Data normalization was performed on the extracted features to ensure the contribution of each feature is equally weighted and this avoids bias that could occur across recording sessions.

3. Experiments

Experiment recording conditions, parameters and constraints are kept consistent across the recording sessions to ensure the reliability and repeatability of the results. Before the start of the recording session, all external device notifications, haptic feedback and audio tone are disabled and the device screen is set to remain active during the experiment. We conduct the experiment in quiet room with relatively low ambient noise less than 50 dB. Both the audio samples and movement data are time synchronised and labelled with the touch input location. In total, 150 touch inputs were collected for each of the nine touch input locations, of which 50 were used to train the classifier with the remaining 100 used to test the model. The training dataset was deliberately limited to 50 touch inputs to avoid over-fitting and to demonstrate that a relatively accurate model can be created with limited inputs. Each session is repeated three times on different days in varying room conditions. This allows us to validate our methodology and ensure the results are consistent across each session.

The movement data and acoustic features extracted from the audio samples are used to distinguish and classify the touch input originating from different locations. To identify the touch input location, a random forest classifier created in Python with scikit-learn [23] is trained on the set of features extracted from the touch input and the results are mapped to one of the nine locations. A random forest classifier with an estimator of 100 trees is selected for this experiment. The choice of the random forest classifier is motivated by the ensemble learning benefits which perturbs-and-combines a number of machine learning models to improve the performance of the classifier. Figure 7 shows the experiment methodology.

Figure 7. Experiment methodology for audio input.

Several experiments were conducted with the movement data and acoustic features extracted from the audio samples. In the first investigation, named *Movement Data Experiment*, movement data is partitioned into five sets with four sets used to train the classifier and one set reserved for tests. The process is repeated five times and the designated test set is rotated in each iteration. In the second investigation, named *Device Orientation Experiment*, the device is relocated and rotated 90 degrees (Figure 8) between the training and test sessions and the movement data is used to identify the touch input location. In the third investigation, named *Acoustic Feature Experiment*, a classifier is trained to identify the touch input location associated with the audio sample. The full touchscreen surface is tested in the experiment and the device relocated and rotated between training and testing sessions to simulate a realistic user-input scenario. Finally, in the fourth investigation, named *Reduced Touch-Input Area Experiment*, the touch input location is reduced and the classifier is trained to evaluate audio samples from the reduced touch input area under realistic conditions where the device is relocated and rotated between training and testing sessions.

3.1. Movement Data Experiment

To investigate using movement data from an Android device to detect touch input, we perform a validation experiment. This experiment also replicates the earlier work [9] used to detect touch input and hence verify our methodology. The device position and orientation are kept constant and the sessions are recorded sequentially to maintain the same device position and orientation across sessions. Touch input from the nine locations are collated into a single dataset and a random forest classifier with 5-fold validation applied is used. In total, 80% of the recorded dataset is allocated for training and the remaining 20% was used to predict the touch input location from the movement data extracted.

3.2. Device Orientation Experiment

In departure from [9] we now include the effects of device position and orientation on the performance and robustness of our model using movement data, under realistic user-input conditions. Training and test datasets are recorded with the device re-orientated between the sessions as seen in Figure 8. Movement data is recorded and feature extraction is performed for all nine touch input locations. A new model is created by applying the random forest classifier to the training data and evaluated against the test dataset.

Figure 8. Overhead view of mobile phone placement showing nine touch input locations with horizontal (**left**) and vertical (**right**) orientation lying on a flat surface.

3.3. Acoustic Feature Experiment

To simulate everyday smartphone usage, we evaluate the efficacy of acoustic features in detecting touch input under field recording conditions. Audio samples for the training dataset are first recorded with the device in a horizontal orientation (Figure 8). The device is relocated, rotated and oriented vertically before recording the test dataset. Segments in the acoustic signal corresponding to touch inputs in the nine locations are isolated and acoustic features described in Section 2.2 are extracted for training and evaluation. A new model is created by applying the random forest classifier to evaluate the performance of acoustic data in identifying touch input.

3.4. Reduced Touch-Input Area Experiment

To better depict everyday usage of mobile phones in vertical orientation, where the keypad area is now reduced to the bottom third of the screen, we now evaluate the performance of acoustic features in distinguishing different touch inputs within a restricted area. Applying similar experimental parameters in *Acoustic Feature Experiment*, touch input is now restricted to 43×66 mm (as opposed to the original 110×66 mm fullscreen area), the effective size of the default number pad input. Audio samples for each touch input location are similarly recorded with the on-board microphones with device position and orientation varied between the training and test sessions. A new model is trained by applying the random forest classifier to the acoustic features extracted from the audio samples in the training dataset. The model is then evaluated against the test dataset to determine the effects of the touch input area on prediction accuracy.

4. Experimental Results

4.1. Movement Data Experiment

The validation experiment successfully identified the touch input location with high accuracy (99%) using the cross-validated movement dataset, similar to the results reported by [9]. The confusion matrix is shown in Figure 9 (correct classification and misclassification indicated in percentage).

To better understand the contribution of the various movement data to the performance of the model, the relative importance of each feature was identified. The weightage of these features contributing to touch input location classification are listed in Table 1, with rotation vector being the most important.

Figure 9. Confusion matrix for Movement Data Experiment.

Table 1. Contribution of movement features in Movement Data Experiment.

Feature	Percentage
R_x	14.8%
R_y	18.4%
R_z	8.0%
R_w	4.6%
L_x	9.6%
L_y	9.5%
L_z	7.7%
G_x	6.8%
G_y	8.6%
G_z	12%

The rotation vector can be understood intuitively as the changes in pitch, roll and yaw of the device when a force is applied on different touch input location of the touchscreen. The rotation vector is represented by the angle of rotation around an axis which align the device's to the current orientation. A force applied near the edges of the device changes its orientation which is recorded as an angle along each axis relative to the reference device orientation. The rotation along all three axes is mapped to a specific region which can then be used alongside the other sensor features together, thus accurately predicting touch input location.

4.2. Device Orientation Experiment

When varying the device position and orientation, we are unable to replicate the findings of *Movement Data Experiment* and the results reported in [9] with our training and test datasets. In contrast with [9], our results show a drastic degradation of prediction accuracy from 99% to 15%, as seen in Figure 10 when the device position and orientation is altered between the training and test session. This suggests that device position and orientation is a key component in the movement data [10] and this reduces the generalisability of the model for other orientations.

Analysing movement data for the training and test session in Figure 11, we observe a separation in the feature space between readings from different sessions for the rotation vector. This suggests that the classification model derived from one session cannot be used to predict touch input location from another session as the data from each session occupies a different feature space. Now, the rotation vector is a composite of various

environment sensors which includes the magnetometer; the magnetometer measures the direction, strength, or relative change of the earth's magnetic field at a particular location and is dependent on the absolute (global) position of the device [24]. Slight changes to the device position will result in a large deviation in the sensor readings. The feature space corresponding to the testing and training session is shown in Figure 11. Readings from the linear acceleration in each axis shown in Figure 11a–c overlap for both training and test sessions. Likewise, gyroscope sensor readings along each axis from the the training and test sessions seen in Figure 11g–i are distributed similarly across the entire range. In contrast, a clear separation between the training and testing sensor readings in all three axes is observed in Figure 11d–f of the rotational vector.

Figure 10. Confusion matrix for Device Orientation Experiment.

Figure 11. Typical histogram representing the sensor data across device orientation. Horizontal range of each plot reflects the range of the points observed, automatically adjusted in scale to maximise the range of values observed; vertical range is the frequency of occurrence. Note the bi-modal distribution for rotation vectors vs. linear acceleration and gyroscope readings suggesting the rotation vector occupies separate feature spaces.

4.3. Acoustic Feature Experiment

The trained classifier successfully predicted if an audio sample originated from a particular touch input location with an average accuracy of 86.2% (Figure 12). The model was able to predict the touch input location with varying degrees of accuracy, achieving best performance near the corners of the touchscreen located near the on-board microphone. This fits with our understanding that using multiple microphones located at different positions allow for increased distinctiveness in acoustic signal attributes. Furthermore the relative position of a particular touch input location to the two microphones remains the same and thus maintains robustness of the acoustic features used for classification irrespective of changes in device position or orientation.

Figure 12. Confusion matrix for Acoustic Feature Experiment.

The relative importance of the audio recognition features from both channels were computed over several experiments and the top common features are listed in Table 2. The classifier (newly created) from these subset of acoustic features resulted in a slight decrease in accuracy, was able to predict touch inputs with an accuracy of up to 82.2%. In the earlier experiment (Section 4.2), device position and orientation certainly matter to the success of the classifier but Section 4.3 show that acoustic features offers resistance to changes in device position and orientation with most of the distinguishing information found in chroma deviation and specific spectral features. The resulting confusion matrix corresponding to the classifier trained using a subset of acoustic features as shown in Figure 13 shows the decrease in accuracy and higher rate of misclassification in neighbouring touch input locations compared to Figure 12.

Table 2. Contribution of acoustic features in Acoustic Feature Experiment.

Feature	Percentage Contribution
Zero Crossing Rate	2.2%
Energy	5.3%
Entropy of Energy	3.3%
Spectral Centroid	0.6%
Spectral Spread	0.2%
Spectral Entropy	2.6%
Spectral Flex	0.6%
Spectral Roll-off	7.0%
MFCC	28.3%
Chroma Vector	49.6%

Figure 13. Confusion matrix for Selected Acoustic Feature experiment.

Acoustic features were further analysed using t-SNE [25]. The resulting t-SNE map that reveals the inter-region heterogeneity and intra-region homogeneity is shown in Figure 14.

In this figure, acoustic features extracted for a given touch input location are colored according to the location label. The t-SNE scatter-plot reveals clear dimensional separations based on acoustic characteristics. Figure 14 also highlights the chances of misclassification of touch input samples in each location with physically adjacent regions having a higher probability of misclassification. Note also a few stray points of red and blue near the cluster of black: this may reflect the possibility of misclassification attributed to chance similarity of the acoustic signals collected.

Figure 14. t-SNE plot for selected acoustic features.

4.4. Reduced Touch-Input Area Experiment

Unsurprisingly, a reduction in the effective touch input area sees a decrease in accuracy with the trained classifier. However, the model is still able to predict the touch input locations with an accuracy of 78.8%, a decrease of just 7.4% from the previous experiment. Touch input locations are now constrained within a smaller area, thereby increasing intra-group variance while decreasing inter-group variability due to increased

touch input proximity. Additionally, touch input locations are now further from the upper microphone and hence suffer from poorer signal-to-noise ratio, thereby reducing distinguishing acoustic features, thus reducing performance as seen from the somewhat higher misclassification rate of the middle row (47% of misclassification for middle row) of touch inputs (cf. Figure 1b) in Figure 15.

Figure 15. Confusion matrix for Reduced Touch Input Area experiment.

5. Discussion

We observe that a cross-validated approach using movement data as reported in [9] resulted in good classification accuracy when distinguishing touch input without changing position and orientation. However, its performance severely degrades when device position and orientation are changed (see Section 4.2), conditions which were not explored by [9] and was in fact overlooked in their investigation. As expected, this limits the efficacy of movement data in locating touch input and motivates seeking pathways which generate sensor data that is resistant to changes in device position and orientation.

Such a pathway which is more robust can be derived by exploiting multiple microphone signals on the devices. The non-similarity of the upper and lower microphone responses to touch input location shown in Figure 3 and fixed positions relative to the device increase the quality of information extracted, while also ensure less sensitivity to changes in position and orientation. Applying audio processing techniques and analysing the acoustic features therein, we can identify the touchscreen location tapped. This approach will yield increased accuracy with additional microphones, as seen in devices such as the recent iPhone models which now include up to four on-board microphones. Previously, techniques such as Time Difference of Arrival (TDoA) analysis have seen application on standard keyboards [7] and touchscreen surfaces [26] to identify possible sets of keys within a restricted area (virtual keyboard) with up to 90% accuracy [27]. In line with the previous studies, acoustic features used in our setup is able to identify touch input location across the entire touchscreen an average accuracy of 86.2%, offering almost comparable performance. These studies reinforce the fact that physical signals arising from user interaction may offer unintended pathways of data compromise.

As introduced in Section 2, MFCC and chroma vectors features contribute over 80% to touch input classification and can be used in combination to create an acoustic fingerprint of different sections of the screen to categorize touch input location, by distilling and removing redundancies in the raw signal and allow analysis to be concentrated on salient attributes of the acoustic event. Comparatively, spectral and energy entropy features contribute only up to 20% to categorise user touch input. Filtering out these peripheral features can improve

the generalisability of our model with only minor degradation in accuracy when applying machine learning techniques to determine user touch input location by identifying acoustic markers.

The use of acoustic features maintains a good discrimination of touch input classes across varying position and orientations, however it is subject to the maximum distance available between touch inputs location; with a smaller touch input area (numberpad input or simply a smaller device), acoustic features extracted may be less distinct due to the increased proximity of user touch input locations which results in a reduction of accuracy—from 86.2% to 78.8%. Even under such limiting conditions, acoustic features still provide comparable performance with detection techniques that rely on accelerometer data to identify touch input over a larger touch input area, with accuracy ranging from 78.0% on smartphones [4] to 84.6% on smartwatches [28] due to the strictly unique signal response and position of each on-board microphone.

This acknowledges the fact that touch input location on touchscreen devices can be retrieved from physical signals associated with the device. It then provides the basis and motivation to further investigate links between the physical characteristics of touch input location, acoustic excitation and how the physical device and signals interact accordingly.

To address the threat of unauthorised data access, techniques to limit fine-grained sensor readings can be employed to complicate keystroke inference. However this approach may pose potential problems for legitimate mobile apps and introduce usability issues as well. A more tempered approach would be to disable access to sensors when user are required to provide input in a sensitive application or include a physical kill-switch. While such potential vulnerability exists whereby an attacker gains access to the microphone sensor on the target device, nevertheless such acoustic-based incursions may nevertheless be minimized by changing the keyboard layout each time, emitting masking sounds which can interfere with touch input location identification or enabling haptic feedback to lower the rate of touch input detection and thus alter the acoustic response used in touch input classification. Unintended data exposure can also be attenuated by changing the mechano-acoustic response of the device to frustrate or invalidate the acoustic fingerprint registered by the trained classifier, such as fitting on a heavy rubber phone cover or dynamically altering the distribution of mass or mechanical coupling on the device. These techniques and counter measures against acoustic incursions are summarised in Table 3.

Table 3. Counter-measures Against Acoustic Incursions.

Category	Counter Measure
Prevention	Limit fine-grained sensor readings Physical kill-switch
Jamming	Alternate keyboard layout Play masking Sounds Enable haptic feedback
Shielding	Add a heavy rubber phone cover Dynamically altering the distribution of mass

6. Conclusions and Future Work

We show it is possible to determine the touch input location on a touchscreen via acoustic and movement information extracted from a mobile device. Acoustic features proved to be more effective under realistic usage conditions compared with movement data alone; user touch input location can indeed be determined from audio recordings using on-board microphone sensors. Thus, the ensemble machine learning algorithm chosen for this investigation is effective in classifying touch input location with an accuracy of 86.2%. This has wide-ranging implications on user input privacy on mobile communication devices armed with on-board mechano-acoustic sensors.

In future work, further investigation of the acoustic signal and physical characteristics of the device will allow us to determine if other acoustic features or features extracted from pre-trained networks will further improve the sensitivity of the model. Neural networks may also be employed to automatically detect touch input. A larger dataset including swipe and long press input across a variety of implements (e.g., stylus) with the device held at different orientation and inclination may also be considered to evaluate the generalisability of the acoustic model. Model transferability across users and devices may be evaluated with the use of pre-trained models created with inputs from different users. This can be further extended with a larger dataset to address dynamic virtual keyboard layouts on different mobile devices such as smartphones and tablets, and collecting additional data with external microphones. This database can be further extended using augmentation techniques. Furthermore, as touchscreen interfaces are adopted by industrial appliances, the approach presented could be used to analyse these devices for potential vulnerability to acoustic side-channels. Finally, the success of incursion mitigation techniques must also be investigated to determine the most effective and practical approaches which can be implemented by users seeking increased data privacy.

Author Contributions: Conceptualization, K.R.T., B.B.T., J.Z. and J.-M.C.; methodology, K.R.T. and B.B.T.; software, K.R.T.; validation, B.B.T.; investigation, K.R.T. and B.B.T.; writing—original draft preparation, K.R.T.; writing—review and editing, B.B.T., J.Z. and J.-M.C.; visualization, K.R.T. and B.B.T.; supervision, B.B.T., J.Z. and J.-M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Barker, E.; Dang, Q. Nist special publication 800-57 part 1, revision 4. *NIST Tech. Rep.* **2016**, *16*, 22–28.
2. Panda, S.; Liu, Y.; Hancke, G.P.; Qureshi, U.M. Behavioral Acoustic Emanations: Attack and Verification of PIN Entry Using Keypress Sounds. *Sensors* **2020**, *20*, 3015. [[CrossRef](#)] [[PubMed](#)]
3. Bucicoiu, M.; Davi, L.; Deaconescu, R.; Sadeghi, A.R. XiOS: Extended application sandboxing on iOS. In Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, Singapore, 14–17 April 2015; pp. 43–54.
4. Owusu, E.; Han, J.; Das, S.; Perrig, A.; Zhang, J. Accessory: password inference using accelerometers on smartphones. In Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications, San Diego, CA, USA, 28–29 February 2012; pp. 1–6.
5. Zhu, T.; Ma, Q.; Zhang, S.; Liu, Y. Context-free attacks using keyboard acoustic emanations. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, 3–7 November 2014; pp. 453–464.
6. Compagno, A.; Conti, M.; Lain, D.; Tsudik, G. Do not Skype & Type! Acoustic Eavesdropping in Voice-Over-IP. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, Abu Dhabi, United Arab Emirates, 2–6 April 2017; pp. 703–715.
7. Zhuang, L.; Zhou, F.; Tygar, J.D. Keyboard acoustic emanations revisited. *ACM Trans. Inf. Syst. Secur. (TISSEC)* **2009**, *13*, 1–26. [[CrossRef](#)]
8. Asonov, D.; Agrawal, R. Keyboard acoustic emanations. In Proceedings of the IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 12–12 May 2004; pp. 3–11.
9. Al-Haiqi, A.; Ismail, M.; Nordin, R. On the best sensor for keystrokes inference attack on android. *Procedia Technol.* **2013**, *11*, 989–995. [[CrossRef](#)]
10. Narain, S.; Sanatinia, A.; Noubir, G. Single-stroke language-agnostic keylogging using stereo-microphones and domain specific machine learning. In Proceedings of the 2014 ACM Conference on Security and Privacy in Wireless & Mobile Networks, Oxford, UK, 23–25 July 2014; pp. 201–212.
11. Lu, L.; Yu, J.; Chen, Y.; Zhu, Y.; Xu, X.; Xue, G.; Li, M. Keylistener: Inferring keystrokes on qwerty keyboard of touch screen through acoustic signals. In Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 775–783.
12. Zhou, M.; Wang, Q.; Yang, J.; Li, Q.; Jiang, P.; Chen, Y.; Wang, Z. Stealing your Android patterns via acoustic signals. *IEEE Trans. Mob. Comput.* **2019**. [[CrossRef](#)]

13. Shumailov, I.; Simon, L.; Yan, J.; Anderson, R. Hearing your touch: A new acoustic side channel on smartphones. *arXiv* **2019**, arXiv:1903.11137.
14. Cheng, P.; Bagci, I.E.; Roedig, U.; Yan, J. SonarSnoop: Active acoustic side-channel attacks. *Int. J. Inf. Secur.* **2019**, *19*, 213–228. [[CrossRef](#)]
15. Cano, P.; Batle, E.; Kalker, T.; Haitisma, J. A review of algorithms for audio fingerprinting. In Proceedings of the 2002 IEEE Workshop on Multimedia Signal Processing, St. Thomas, VI, USA, 9–11 December 2002; pp. 169–173.
16. Wang, A. An Industrial Strength Audio Search Algorithm. In Proceedings of the ISMIR 2003: Proceedings of the fourth International Conference on Music Information Retrieval, Baltimore, MD, USA, 26–30 October 2003; pp. 7–13.
17. Garcés, M.; Fee, D.; Steffke, A.; McCormack, D.; Servranckx, R.; Bass, H.; Hetzer, C.; Hedlin, M.; Matoza, R.; Yepes, H.; et al. Capturing the acoustic fingerprint of stratospheric ash injection. *Eos Trans. Am. Geophys. Union* **2008**, *89*, 377–378. [[CrossRef](#)]
18. Teo, K.R.; Balamurali, B.T.; Chen, J.m.; Zhou, J.Y. Retrieving Input from Touch Interfaces via Acoustic Emanations. In Proceedings of the 2021 IEEE Conference on Dependable and Secure Computing, Aizuwakamatsu, Japan, 30 January–2 February 2021.
19. Culurciello, E. e-Lab VideoSensors. 2017. Available online: <https://github.com/e-lab/VideoSensors> (accessed on 13 October 2017).
20. Giannakopoulos, T. pyaudioanalysis: An open-source python library for audio signal analysis. *PLoS ONE* **2015**, *10*, e0144610. [[CrossRef](#)] [[PubMed](#)]
21. Teo, K.R.; Balamurali, B.T.; Ng, T.S.; Chen, J.m. Exploring DiZi Performance Parameters With Machine Learning. In Proceedings of the 2018 ACM Conference of the Australasian Computer Music Association, Perth, Australia, 6–9 December 2018.
22. Shepard, R.N. Circularity in judgments of relative pitch. *J. Acoust. Soc. Am.* **1964**, *36*, 2346–2353. [[CrossRef](#)]
23. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
24. Semiconductors, N. Xtrinsic MAG3110 Three-Axis, Digital Magnetometer. 2013. Available online: <https://www.nxp.com/docs/en/data-sheet/MAG3110.pdf> (accessed on 2 February 2017).
25. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
26. Miluzzo, E.; Varshavsky, A.; Balakrishnan, S.; Choudhury, R.R. Tappprints: Your finger taps have fingerprints. In Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, Lake District, UK, 26–29 June 2012; pp. 323–336.
27. Gupta, H.; Sural, S.; Atluri, V.; Vaidya, J. Deciphering text from touchscreen key taps. In Proceedings of the IFIP Annual Conference on Data and Applications Security and Privacy (DBSec), Trento, Italy, 18–20 July 2016; pp. 3–18.
28. Maiti, A.; Jadliwala, M.; He, J.; Bilogrevic, I. Side-channel inference attacks on mobile keypads using smartwatches. *IEEE Trans. Mob. Comput.* **2018**, *17*, 2180–2194. [[CrossRef](#)]