Evaluating potential sensitive information leaks on a smartphone using the magnetometer and Conformal Prediction

Robert Choudhury Robert.Choudhury.2015@live.rhul.ac.uk
Zhiyuan Luo Zhiyuan.Luo@rhul.ac.uk
Khuong An Nguyen Khuong.Nguyen@rhul.ac.uk
Centre for Reliable Machine Learning, Royal Holloway University of London, Surrey, TW20 0EX, United Kingdom

Editor: Harris Papadopoulos, Khuong An Nguyen, Henrik Boström and Lars Carlsson

Abstract

The low powered sensors used in modern Smartphones do not require permissions when using low sampling rates i.e. 200Hz and below. This has made them a target for side channel attacks. In this paper we perform a series of experiments that harvest raw data from the low powered sensor known as the magnetometer. We start by using unsupervised learning with the cosine metric to provide clear indications if it is possible to classify the data into the different security events occurring at the time of capture. We then build a model, designed to be robust in terms of the orientation of the device, to evaluate the risk of sensitive data being correctly identified from magnetometer data despite the limited sampling rate. Using a model trained with LSTM on the whole data set with an 80/20 split, our results show 100% accuracy on our reverse Turing test and 67.5% on the key press test. We also show that when analysing the captured magnetometer responses to playing sound samples from the loudspeaker it is very difficult to infer the original sound. We extend the work using Inductive Conformal Prediction by examining the property of uncertainty for different confidence levels. We also show that despite a high degree of uncertainty there is the potential to infer security properties such as the layout of a screen. To this end we show that the number 5 in the center of a keypad occurs a disproportionately high number of times in the prediction set (68.3%).

Keywords: Conformal Prediction, Reverse Turing Test, Mobile Security

1. Introduction

Smartphones are part of our daily lives, they allow users to conduct online transactions, take photos and even play games. To support and enrich the user experience, sensors have been added to enable the detection of the context of a user's interaction with the phone. For example, by detecting the phone's orientation through the sensor readings, app developers may adjust the screen to portrait or landscape for a better viewing experience; or switch the screen off to avoid unintended touches when the phone is near the face using the proximity sensor; or adjust the screen brightness using the ambient light sensor.

However, despite being widely used in many mobile apps, these low powered sensors do not require any permissions from the user and are potential targets for side channel attacks. Malicious actors can design apps that harvest the sensor readings to infer the user's activities. Detecting such malicious use is a difficult task for Machine Learning and

often results in many unwanted false positives. Therefore, in this paper, we implement a combination of unsupervised learning, deep learning and Conformal Prediction to detect potentially sensitive information being leaked via the magnetometer. We assess the validity and accuracy of our algorithm in three different real-world scenarios as follows:

- Detecting if a phone has been deliberately cut off from external signals (Reverse Turing test).
- Eavesdropping onscreen typing by inference.
- Examining possible leakage of sensitive information from the phone's loudspeaker.

1.1. Paper's contributions

This work makes the following contributions:

- Demonstrates that the magnetometer on a smartphone is sensitive enough to allow the construction of a reverse Turing test that can be used as a trigger for offensive and defensive security actions.
- Introduces the use of a confidence measure to aid in identifying and comparing potential leaks of sensitive information from low powered sensors.
- Uses unsupervised learning techniques to visualise the noisy magnetometer data to highlight the potential risk of sensitive information being leaked.

1.2. Structure of the paper

The rest of this paper is structured as follows. Section 2 gives an overview of relevant work in this field. Section 3 provides the motivation for this work. Section 4 describes the methodology for obtaining the data from our apps. Section 5 analyses and discusses the results from the models created. Section 6 concludes our work and discusses future work.

2. Related work

It is possible for a malicious actor to circumvent the security policy of a smartphone by inferring sensitive information using low powered sensors such as accelerometers. Nguyen et al. demonstrated this by using magnetometer and accelerometer traces to track the movement of a target's smartphone (Nguyen et al., 2019, 2017). It was also demonstrated that JavaScript and a locally installed app could, with only 100 sensor samples, infer the device factory calibration and allow fingerprinting of a device across multiple platforms (Zhang et al., 2019; Amerini et al., 2017).

Keystroke inference is the process of identifying the keystrokes that have been made by an unsuspecting user. In a machine learning context this involves the process of capturing many samples and preprocessing the data into a format that is used to build a model. When a user enters information on the phone screen an eavesdropper uses keystroke inference by capturing the sensor readings, performing the relevant sensor preprocessing and using the

model to classify which keystroke has most likely been made (Javed et al., 2020; Giuffrida et al., 2014).

Eavesdropping without the use of a machine learning model has been achieved in MagEar (Bulusu et al., 2022), where the author designed a custom device that captures magnetic signals using a custom coil with a high enough sensitivity to detect the changes produced by a transducer in the victim's headphones. The captured and processed magnetic signals can be listened to and placed into speech recognition software with a high degree of accuracy. Our research differs because we are limited to the use of the onboard sensors that are available as part of a typical smartphone. These do not possess the sensitivity to achieve comparable results yet.

An empirical study to fingerprint publicly available malware analysis services was conducted in (Botas et al., 2018). Samples were sent to each platform and artifacts such as the version of the operating system and the MAC address were retrieved. As many of these values were shown to be common or the same on various analysis platforms, the authors showed it is possible to fingerprint analysis environments using these values. A method was proposed to prevent fingerprinting by generating a random value for each of the artifacts which was then fixed and returned to the querying sample. This differs from our work which is focused on the mobile operating system Android and more specifically the returned values from sensors. This work was extended in Choudhury et al. (2022) where an attack was proposed that would defeat the random artifacts framework if applied to sensor readings produced by mobile devices.

In the paper 'Tap Wave Rub' (Shrestha et al., 2015) the authors produced a Reverse Turing test based around the sensor readings recorded when the user was prompted to perform a sequence of uncommon gestures to ensure that near field communications (NFC) were correctly triggered by the human user and not by malicious software installed on the device. This work has the benefit of being able to detect an attack in real time and not posteriori. In 2019 TrendMicro analysed two apps namely Batterysavermobi and Currency Convertor which both use a threshold of the accelerometer value as a means to detect if the malicious app is under investigation (Sun, 2019).

3. Background of sensor security on Android

In this section we provide an overview of the background of sensor processing on Android and potential security implications.

3.1. Low powered sensors access on Android

Low-powered sensors are energy efficient sensors designed to be used to collect information that can be processed in a running app to improve the end user experience or add additional features. The information produced by these sensors is not considered to be sensitive and therefore app developers do not require permissions to gain access to this information. Some examples of low-powered sensors that may be contained in a modern smartphone are:

- Accelerometer: measures the acceleration of the device in three axes and the phone's tilt.
- Gyroscope: measures angular velocity and the orientation of the phone.

- Proximity sensor: detects the presence of nearby objects without physical contact.
- Ambient light sensor: measures the amount of light in the surrounding environment
- Magnetometer: measures the direction and strength of a magnetic field in three axes.

The Android documentation team have published on their website (Android Developers, 2023) that by using the permissions model in the recommended way the following security objectives should be fulfilled:

- Control - a user retains control over any data shared with other applications.
- Transparency - a user understands what data an app is using and why.
- Data minimization - the data used is required for a specific task or invoked action.

Permissions are further divided into run time and install time. Run time permissions are where more dangerous permissions such as access to the microphone and camera are acquired with the user being prompted when they are needed. Install time permissions are declared in a manifest file and are for access to non sensitive resources such as the ability to access the internet.

3.2. The reverse Turing test

The Turing test was named after Alan Turing and involves an interrogator querying a subject to determine if it is a computer or a human. Conversely a reverse Turing test is a computer program trying to determine if it is interacting with a human operator.

In the case of evasive malware, the program is trying to determine if inputs provided by a device are from a human operator or are inputs provided by a malware analysis platform. Malware can perform the test by observing real-time interactions between a human user and a device for example by prompting the user to perform a task such as clicking a button and looking at accumulated wear and tear that occurs through usage of a live system. A real world example of a positive use of a reverse Turing test is CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart). This is used to challenge a user for a response that only a human can perform.

In this research an experiment is designed to test if a phone has had its signal deliberately blocked thereby disabling the ability of the device to communicate. There are situations where this might be desirable such as an investigator trying to prevent evidence from being tampered with remotely or a journalist trying to ensure that their phone's location is not broadcast when they are meeting a vulnerable source. This is a reverse Turing test because a program on the device is trying to ascertain if the sensor data is from a phone that is still in the possession of its user. In this context the test can be used to trigger routines on a device such as starting to record using the microphone or storing the last known GPS location. In order to investigate if this reverse Turing test is possible we built a computer model to classify if the sensor data being generated from a phone is from a handset that is either placed inside a signal blocking bag or outside.

3.3. Threat model

We assume that the data gathering app is installed on the target device, allowing it to gather readings directly from the device's sensors. This is considered a reasonable assumption as the use of low powered sensors is ubiquitous and does not need permissions unless the app requires a higher sampling rate than 200Hz (Google, 2022).

4. Detecting information leaks from a magnetometer

In this section we present our proposal on the use of unsupervised learning and the application of Inductive Conformal Prediction to detect information leakage on smartphones.

4.1. Processing magnetometer readings

The magnetometer is a low powered sensor designed to measure the magnetic field strength around a mobile device. Examples of its use include compasses and tracking direction that the device is facing / travelling in.

Magnetometers consist of three sensors, one for each axis of the phone, and work by detecting the strength of the magnetic field along each axis(see Figure 1). These readings are dependent on the orientation of the sensor and therefore must be processed or alternatively must be taken with the phone in a specific position. Some mobile devices contain uncalibrated and calibrated sensors. Because every sensor will return slightly different readings due to variations in the hardware, sensors are calibrated to try to remove any potential bias that may hinder performance. The sensors used in the test devices were calibrated.

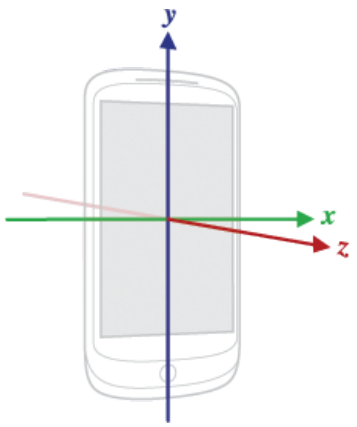


Figure 1: A diagram of the axes (Google, 2022).

Formally raw data consists of i time sequence of tuples:

$$(t_i, x_i, y_i, z_i), i = 1, ..., N$$

where t_i is the date/time of the sensor event, and x_i, y_i, z_i are the values returned from the sensor at the corresponding axis and N is the number of times the sensor event has occurred as defined by the `onSensorChanged()` method being called.

When capturing the data, to allow for independence of orientation, the total magnetic flux density is calculated using the following formula:

$$tm_i = \sqrt{(x_i^2 + y_i^2 + z_i^2)} \tag{1}$$

Although this process aids with producing a robust model there is the potential issue that extremely small starting values will become smaller decreasing the signal to noise ratio, therefore making it harder to classify the captured event.

4.2. Unsupervised learning: visualising the collected sensor readings

In this research we use the dimensionality reduction algorithms UMAP (Uniform Manifold Approximation and Projection) and t-SNE (t-distributed Stochastic Neighbor Embedding). Both algorithms attempt to preserve the global structure of the data whilst reducing the dimensionality. We are specifically using these algorithms for their ability to group similar data points together into clusters in a 2 and 3-dimensional space. In particular UMAP was chosen to reduce the computational complexity and time taken to produce the graph.

4.2.1. Cosine similarity

In conjunction with UMAP the cosine similarity metric was chosen because magnetic field strength is affected by a high degree of background noise, potentially leading to irrelevant features.

As cosine similarity considers the angle between the vectors being compared and not their respective size we believed that it would perform better in the presence of background noise such as the Earth's magnetic field.

Mathematically the cosine similarity (Manning et al., 2008) between two vectors a and b is calculated as follows:

$$\text{Cosine similarity}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{||\mathbf{a}||||\mathbf{b}||} \tag{2}$$

Here, $\mathbf{a} \cdot \mathbf{b}$ is the dot product between the two vectors and $||\mathbf{a}||$ and $||\mathbf{b}||$ are the norms (magnitudes) of each vector. The cosine similarity measures the cosine of the angle between the two vectors and is a value between -1 and 1 where 1 means the two vectors are identical, 0 means they are orthogonal and -1 means they are opposite.

4.3. Inductive Conformal Prediction for data leakage detection

Conformal Prediction (Vovk et al., 2005) has value in fields where errors in prediction can lead to unknown or high consequences such as in medicine, where it can be used to aid doctors in their disease diagnosis. By examining confidence intervals a doctor can gain insight into the level of uncertainty there is in a model's prediction. Conformal prediction has been used in a security context to provide confidence in malware classification as an aid to existing signature and heuristic based malware detection platforms so that an analyst has a statistically sound measure of how likely an Android application is to be malicious Georgiou et al. (2016).

In this paper Inductive Conformal Prediction (ICP) is used to provide the concept of confidence levels in the model's predictions (Vovk et al., 2005; Vovk, 2012). Inductive Conformal Prediction has been used in our research because the underlying model does not need to be retrained for every new test sample. This makes it useful for any model that is expensive to train.

Prediction consists of two main steps: calibration and prediction. Given a training data set of n samples $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ where x_i is a feature vector and y_i is the corresponding label. Sample, x_i represents a sensor reading from a magnetometer and y_i is the sensitive information being generated for the experiment. For a new input x_{n+1}, we aim to predict the label $\hat{y}_{n+1}$ at a certain confidence level.

The training set is spilt appropriately into a proper training set $\{(x_1, y_1), \ldots, (x_k, y_k)\}$ of size k and a calibration set $\{(x_{k+1}, y_{k+1}), \ldots, (x_n, y_n)\}$ of size $n-k$. The proper training set is used to build a model.

For each candidate label, y^c for the new input x_{n+1}, compute the conformity score α_i, for each sample in the calibration set and α^c_{n+1} for the new input:

$$\alpha_i = A(\{(x_{k+1}, y_{k+1}), \ldots, (x_{i-1}, y_{i-1}), (x_{i+1}, y_{i+1}), \ldots, (x_n, y_n), (x_{n+1}, y^c)\}, (x_i, y_i)), \quad i = k+1, \ldots, n \tag{3}$$

$$\alpha^c_{n+1} = A(\{(x_{k+1}, y_{k+1}), \ldots, (x_n, y_n)\}, (x_{n+1}, y^c)) \tag{4}$$

where A is a conformity measure typically defined using an underlying algorithm trained on the proper training set, a function that quantifies the conformity of an example given a bag of other examples in the calibration set and (x_{n+1}, y^c).

Based on those conformity scores, the p-value for each candidate label y^c is calculated:

$$p(y^c) = \frac{|\{i = k+1, ..., n : \alpha_i \leq \alpha^c_{n+1}\}| + 1}{n - k + 1} \tag{5}$$

Finally, the prediction set for the new input x_{n+1} can be constructed for a specified confidence level $1 - \epsilon$:

$$\Gamma_{1-\epsilon}(x_{n+1}) = \{y^c : p(y^c) > \epsilon\} \tag{6}$$

Note that the prediction set $\Gamma_{1-\epsilon}$ may contain the empty set, a single label or multiple labels.

4.3.1. Design

The underlying model used was Long Short Term Memory (LSTM). This model was chosen because it is relatively efficient to train and delivered consistent results with regard to accuracy and it is designed to handle sequences. The design of LSTM allows it to remember key features across multiple time steps. This architecture is also designed to help overcome the problem of vanishing gradients which, due to the length of the data sequence (over 500 time steps), is an issue. Transformers were also implemented, however we found that there was a large increase in execution time and the increase in accuracy for our data sets was negligible. The Random Forest algorithm was also implemented but we found that it was inconsistent with respect to accuracy across multiple executions .

5. Empirical results

Following the outline of the principals behind our methodology, this paper continues by displaying the visualisations from the unsupervised learning techniques used and analysing the results gathered against the data collected.

5.1. Experiment scenarios

In this section we describe the details of a series of experiments using Android sensors. The goal is to see if sensitive information can be inferred from the readings obtained from low powered sensors specifically the magnetometer.

5.1.1. Experiment 1: Detecting if a phone has been deliberately cut off from signal (Reverse Turing test)

In this experiment a Faraday bag was used as a signal-blocking bag. A Faraday bag is a portable case that is lined with a conductive material that shields the internal device from external electromagnetic signals and stops the internal device's emissions from escaping. The test phone was placed inside the bag and readings were taken. The phone was removed from the bag and readings were collected from three other locations. This problem is treated as a binary classification problem: 'in the bag or out' where class 0 is inside the bag and class 1 outside.

5.1.2. Experiment 2: Inferring typing on a touchscreen

Most smartphone touchscreens are manufactured using a thin layer of a transparent conductor such as indium tin oxide. They detect touch by detecting the change in capacitance caused when a human finger (or any natural conductor) makes contact with the screen. This change in capacitance was measured across a grid that represents the phone's screen. For this experiment a user pressed on a numbered keypad and the magnetometer readings were recorded. For this experiment we focused on 3 classes namely the number keys 1, 5, 9 which are treated as classes 0,1,2 respectively.

5.1.3. Experiment 3: Examine possible leakage of sensitive information from the internal speaker

In this experiment the loudspeaker of the phone was used to play the audio file for the required number of captures. An app was installed on the victim's device to capture the changes in the sensor readings and to write the results to a file which was later retrieved and processed. The audio files are human spoken numbers from 1 to 10, and are mapped to the classes 0 to 9 in ascending numeric order.

5.2. Data set capture

A series of mono sounds was produced specifically for this research so they would not be subject to copyright laws. The first set consisted of a continuous 1 kHz sine wave tone lasting 3 seconds and a series of 1 kHz 25 ms bleeps. The tone and bleeps were chosen as they represent extremes in audio so they would be easier to classify. These were used for

the purpose of binary classification. The second set was a set of numbers from one to ten recorded by a female voice. Each voice recording file was the same length and was processed with dynamic range compression to ensure the same range of signal amplitude. This meant that they are difficult to classify when compared to a random selection of human speech. Numbers were chosen as they are often used to provide one time passwords as a way of authenticating a user.

A Ulefone 9P Note with Android 10 installed was used initially to capture readings, see Figure 2. An app was developed and loaded on this device. It was designed to start recording sensor values and then play the sound after a set delay to allow time for the file to buffer. This set delay also means that all experiments will have a consistent delay prior to the audio starting. The app has been developed through several iterations to allow multiple sensitive events to be captured and multiple captures to be quickly made. The app's interface was also later redesigned to allow for the capture of different types of sensitive events such as key presses by adding a keypad.

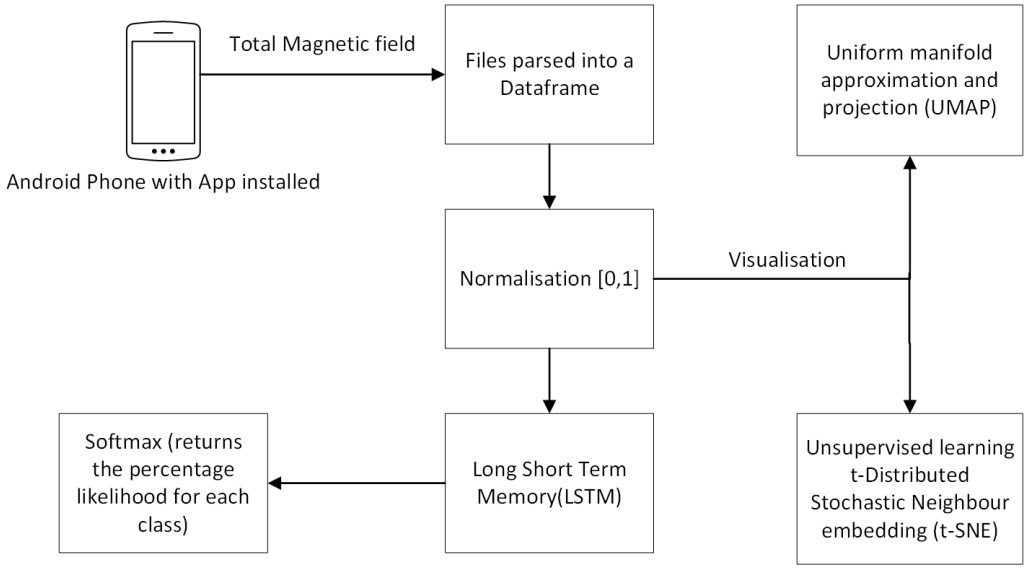


Figure 2: System used to capture and analyse data

A set of sensor events was produced such that:

$TA = ta_i^s, ..., ta_i^e,\ i = 1, ..., N$ where s is the start of the capture and e is the end and i is the place in the sequence of captured sensor readings.

These captures consist of the tuples: (TA, L_i) where L_i is the label of the sensitive event occurring during TA.

Structure of the captured data sets For scenario #1 our collected data set consisted of 2,000 captures between two classes, for scenario #2 our data set consisted of 600 captures spread between 3 classes and finally scenario #3 had 10,000 captures spread between 10 classes. In all cases the class distribution was uniform.

Processing of the data was performed using the `Scikit-learn` package (Pedregosa et al., 2012) and ICP was implemented based on the `nonconformist` API with the margin error

function as the non conformity measure $0.5 - (\hat{P}(y_i|x_{n+1}) - max_{y^c != y_i}\hat{P}(y^c|x_{n+1})) * 0.5$ for each candidate label y^c for the new input x_{n+1} (Linusson, 2014).

5.3. Evaluation criteria

To identify the risks of a leak from raw data of a 'sensitive' event we created a training data set labelled with the sensitive event and started with dimensionality reduction to highlight any potential relationships. A base classifier was trained three times using LSTM with a split of 80%/20% for training and test data respectively, the accuracy values were averaged and recorded below:

- Experiment 1 accuracy 100%
- Experiment 2 accuracy 67.5%
- Experiment 3 accuracy 22.57%

Conformal Prediction was used to add the three additional measures of Emptiness, Uncertainty and Validity for a given confidence level.

- **Emptiness** occurs when the conformal predictor rejects all the possible labels for a test sample for a given confidence level. In terms of security this is a desirable property given that the sensor data gathered should be difficult to classify as a piece of potentially sensitive information. Emptiness gives a measure of this difficulty that would not be present with a normal classifier which chooses the label with the highest probability of being correct according to its model.
- **Uncertainty** occurs when the conformal predictor returns multiple labels for a test sample at a given confidence level. This suggests that the model is having difficulty differentiating between the labels. Depending on the security context, this information can infer some details about the labels, not just that it is difficult to differentiate between them but that the labels share a relationship. In the example of the spatial relationship of a keypad, if you have a tight prediction region with high confidence and samples include predictions from two labels, this may suggest that these items are physically close to each other if the captured values are affected by their spatial relationship.
- **Validity** measures the probability of the true label falling into the predicted region with the specified level of confidence. This means that as samples are analysed, the proportion of predictions that do not contain the correct label should not exceed the specified significance level.

5.4. Preprocessing

To optimize the performance of the deep learning algorithm when working with a raw sequence, the data must be processed such that $tm_{std} \in \mathbb{R} : 0 \leq tm_{std} \leq 1$ where tm is the total magnetic reading which is given by the formula (**??**).

Noise in the total acceleration data is reduced and the data is scaled to meet this requirement using the MinMax scaler algorithm:

$$ta_{std} = \frac{ta - ta_{min}}{ta_{max} - ta_{min}} \tag{7}$$

Magnetometers are known to have a low sampling rate when compared to other sensors. Because we were using a phone with Android version 10 we decided to check if the recently implemented sensor update rate limit (Google (2022)) applied to a slightly older phone. We computed the sampling rate of the sensor using the following formula on our sampling files:

The first step was to calculate how much time had elapsed during the running of multiple files by taking the start time and end time of each file.

$$\text{Average Time Elapsed } (\mu) = \frac{\sum_{i=1}^{N}(t_i^{\text{end}} - t_i^{\text{start}})}{N} \tag{8}$$

In this formula, N is the number of time pairs, t_i^{start} is the start time of the ith pair, and t_i^{end} is the end time of the ith pair.

The second step was to calculate the number of sensor updates. Each line in each capture file represents one sensor update.

Therefore the average number of sensor updates was found by counting how many sensor updates were written to each file and taking an average.

$$\text{Sampling rate} = \frac{\text{average number of sensor updates}}{\mu} \tag{9}$$

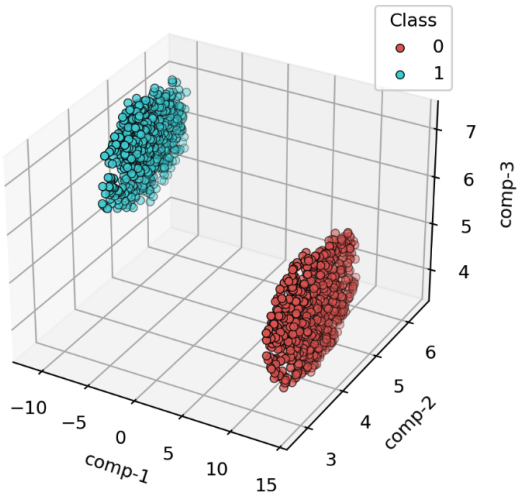
During Experiment 3 the sampling frequency of the magnetometer was 200.00 Hz when recording the playback of a series of numbers. The other experiments had very similar sampling rates, with Experiment 1 having a rate of 199.41 Hz and Experiment 3 200.51 Hz. We can therefore conclude that the sampling rate limit has been implemented on our test phone. These sampling frequencies are much lower than those used in audio sources that are commonly encountered in day to day life. An example is 'telephone audio' which has a bandwidth of 4khz and a sampling frequency of 8 kHz (Rabiner and Schafer, 2007).

5.5. Visualisations

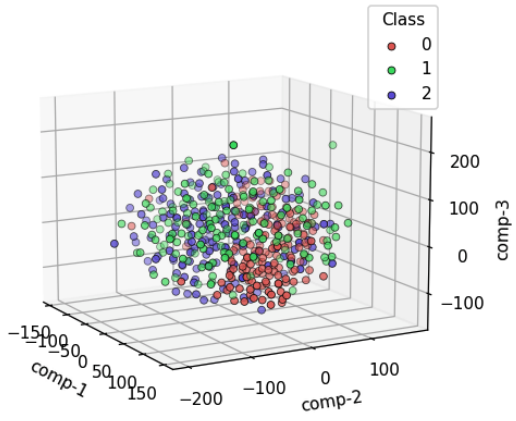
In this section we explore the first method of looking at possible information leakage by using dimensionality reduction algorithms on the data and looking for clusters that correspond to the classes.

In Figure 3(a) we see a tight grouping with clear separation of both the classes which suggests that the cosine metric is working well for this data set. It also suggests that it should be possible for a classifier to work well. We also produced a t-SNE plot with default settings for comparison but there was not a clear separation. We believe this was due to the choice of metric.

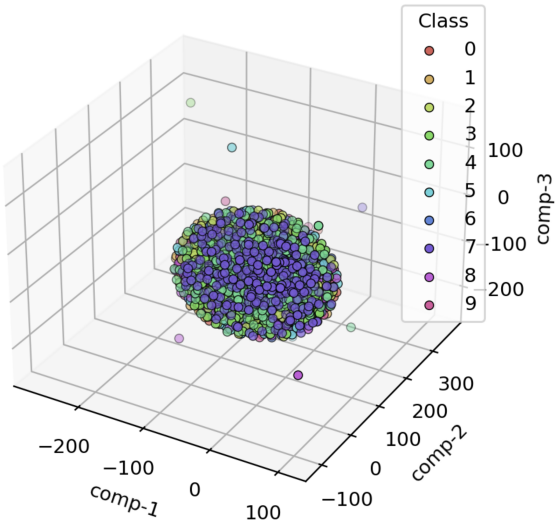
In Figures 3(b) and 3(c) although there is clustering there is no clear separation of the classes which could indicate that the conformal predictor will show a higher degree of uncertainty.



(*a*) Experiment 1: Reverse Turing test.



(*b*) Experiment 2: keypad usage.



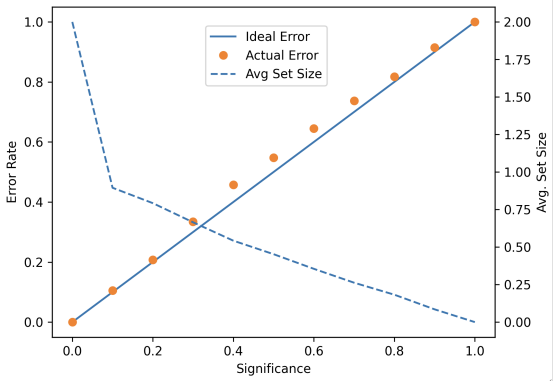
(*c*) Experiment 3: internal speaker playback of numbers.

Figure 3: Dimensionality reduction of magnetometer changes.

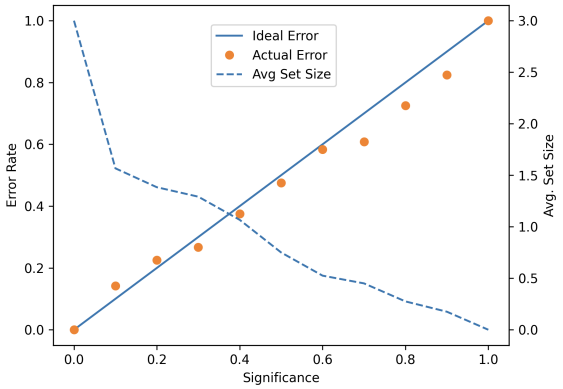
5.5.1. Analysis of calibration curves

As previously stated, the significance level is the maximum level / proportion of incorrect predictions allowed by the conformal predictor. In Figure 4(*a*) the curve follows the ideal $y = x$ line showing that the model is well calibrated. The dashed line on all of the figures shows the average set size vs the significance level. This shows that as the significance level increases the average number of predictions decreases (in binary classification each prediction set size can be either 0, 1 or 2) which is as expected.

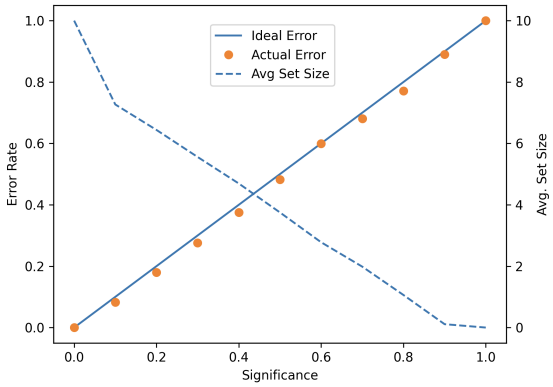
In Experiments 2 and 3 the calibration curve Figures 4(*b*) and 4(*c*) follows the ideal $y = x$ line. As with Experiment 1 it shows that the model has been calibrated correctly.



(*a*) Experiment #1.



(*b*) Experiment #2.



(*c*) Experiment #3.

Figure 4: Example Calibration curve of the 3 experiments with ICP.

5.5.2. Performance analysis with Inductive Conformal Prediction and the security implications

In this section we look at the performance of ICP on the data sets produced by our experiments. ICP performance results across experiments at the confidence level of 95%, 90% and 85% are shown in Tables 1(*a*), 1(*b*) and 1(*c*). These results present the average of three repeated experiments. The training data was further split from the standard 80/20 split for the LSTM model down to 48/32/20 for the training, calibration and test sets respectively.

Experiment 1 performs well as expected given the clear separation of the classes in Figure 3(*a*). This is reflected in the fact that in the test set there is zero uncertainty. We believe the reason for this clear separation is the high concentration of magnetic materials designed to deflect signals in a Faraday bag. In a security context this is undesirable as apps could be designed to detect this change in circumstance and then trigger an event

such as turning on a secondary tracking mechanism for example recording accelerometer movements or the output of the microphone. A real world example is Chelsea Manning who was convicted of violations of the espionage act asked visitors to place their phones inside a disused microwave which acted as a Faraday cage to prevent eavesdropping on interviews after her release Shaer (2017).

In Experiment 2 there is relatively high accuracy as shown in the tables, however the uncertainty is high which shows that the model is finding it difficult to differentiate between classes. Uncertainty in this context could be used to show that the keys are relatively close together.

In Experiment 3, classifying the magnetic field changes caused by the internal speaker proved to be very challenging. We believe the low level of accuracy is in part due to the use of sounds which are the same volume and the same length which is not typical of spoken language. Also we believe that the position of the sensor relative to the speaker is a factor. The conformal predictor also confirms that the classifier is having difficulty distinguishing between classes with the value of uncertainty at over 88 percent.

Table 1: ICP performance across experiments at different confidence levels.

(a) 95% confidence level.

Experiment	**Uncertainty**	**Emptiness**	**Error rate**
Experiment 1 Reverse Turing test	0%	5.0%	5.0%
Experiment 2 Keypad presses	64.17%	0%	6.67%
Experiment 3 Internal speaker	88.90%	0%	4.36%

(b) 90% confidence level.

Experiment	**Uncertainty**	**Emptiness**	**Error rate**
Experiment 1 Reverse Turing test	0%	9.25%	9.25%
Experiment 2 Keypad presses	55.83%	0%	10.83%
Experiment 3 Internal speaker	88.90%	0%	8.60%

(c) 85% confidence level.

Experiment	**Uncertainty**	**Emptiness**	**Error rate**
Experiment 1 Reverse Turing test	0%	15.50%	15.50%
Experiment 2 Keypad presses	49.17%	0%	15.83%
Experiment 3 Internal speaker	88.90%	0%	12.97%

Information leakage via Uncertainty It maybe possible to infer a keyboard layout using the property of uncertainty. We demonstrate this is a potential attack against high security entities that randomise their keyboard layout for the entry of security codes (such as one time pass codes) by making use of data captured when pressing numbers on a keypad layout such as Figure 5.

The number 5 is in the center of the keypad which means that if there is a relationship between spatial layout of the keypad and uncertainty, then the number 5 should feature heavily in the prediction intervals from the conformal predictor. We tested this theory and found that at a confidence level of 85 percent the number 5 returned true 68.33 percent of the time.

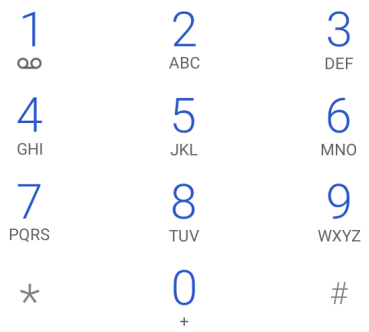


Figure 5: Ulephone keypad.

6. Conclusion and further work

We have presented a novel approach to detecting the information leaks via a magnetometer using raw data and confidence algorithms from Conformal Prediction. We have shown that because the magnetometer has an extremely low sampling rate (0.025 percent of the sampling rate used for telephone audio), it presents a challenge when trying to distinguish between information that requires high fidelity such as classifying different audio played through the internal speaker. We identified other challenging characteristics which are specific to the magnetometer such as the high degree of background noise, and implemented methods to decrease its impact. To that end we made novel use of LSTM in order to work directly with the data instead of extensive feature engineering. We hope that this approach will allow the system to be more portable to other sensors on mobile phones.

We developed a model that can, with 100% accuracy, conduct a form of reverse Turing test by detecting if a phone has been placed in a Faraday bag and is therefore not a live device. We identified scenarios where a hostile actor may use this, such as identifying when a journalist is meeting a vulnerable source.

As further work we would like to incorporate more sensors so that the models' performance can be compared and we can determine which low powered sensors present the highest risk to sensitive information. We also intend to extend the paper to use the higher rates of sampling available via permissions. The results can then be compared to see how effective the sampling rate permissions are in limiting attacks.

Acknowledgements

The work of Robert Choudhury was supported by the EPSRC and the UK Government as part of the Centre for Doctoral Training in Cyber Security at Royal Holloway, University of London (EP/P009301/1).

References

Irene Amerini, Rudy Becarelli, Roberto Caldelli, Alessio Melani, and Moreno Niccolai. Smartphone fingerprinting combining features of on-board sensors. *IEEE Transactions on Information Forensics and Security*, 12(10):2457–2466, 2017.

Android Developers. Permissions on android, 2023. URL https://developer.android.com/guide/topics/permissions/overview.

Álvaro Botas, Ricardo J. Rodríguez, Vicente Matellán, and Juan F. García. Empirical study to fingerprint public malware analysis services. In Hilde Pérez García, Javier Alfonso-Cendón, Lidia Sánchez González, Héctor Quintián, and Emilio Corchado, editors, *International Joint Conference SOCO'17-CISIS'17-ICEUTE'17 León, Spain, September 6–8, 2017, Proceeding*, pages 589–599, Cham, 2018. Springer International Publishing. ISBN 978-3-319-67180-2.

Nirupama Bulusu, Ehsan Aryafar, Aruna Balasubramanian, Junehwa Song, Qianru Liao, Yongzhi Huang, Yandao Huang, Yuheng Zhong, Huitong Jin, and Kaishun Wu. MagEar: eavesdropping via audio recovery using magnetic side channel. *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, pages 371–383, 2022. doi: 10.1145/3498361.3538921.

Robert Choudhury, Zhiyuan Luo, and Khuong Nguyen. Malware in Motion. *Proceedings of the 8th International Conference on Information Systems Security and Privacy*, pages 595–602, 1 2022. doi: 10.5220/0010976200003120.

Nestoras Georgiou, Andreas Konstantinidis, and Harris Papadopoulos. Malware Detection with Confidence Guarantees on Android Devices. *IFIP Advances in Information and Communication Technology*, pages 407–418, 1 2016. ISSN 1868-4238. doi: 10.1007/978-3-319-44944-9_35.

Cristiano Giuffrida, Kamil Majdanik, Mauro Conti, and Herbert Bos. I sensed it was you: authenticating mobile users with sensor-enhanced keystroke dynamics. In *Detection of Intrusions and Malware, and Vulnerability Assessment: 11th International Conference, DIMVA 2014, Egham, UK, July 10-11, 2014. Proceedings 11*, pages 92–111. Springer, 2014.

Google. Sensors Overview — Android Developers, 2022. URL https://developer.android.com/guide/topics/sensors/sensors_overview.

Abdul Rehman Javed, Mirza Omer Beg, Muhammad Asim, Thar Baker, and Ali Hilal Al-Bayatti. Alphalogger: Detecting motion-based side-channel attack using smartphone keystrokes. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–14, 2020.

Henrick Linusson. GitHub - donlnz/nonconformist: Python implementation of the conformal prediction framework., 2014. URL https://github.com/donlnz/nonconformist. Created by Henrik Linusson this is a library that allows the use of Inductive and Transductive conformal prediction.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *An Introduction to Information Retrieval*. Cambridge University Press, 1st edition, 2008.

Khuong An Nguyen, Chris Watkins, and Zhiyuan Luo. Co-location epidemic tracking on london public transports using low power mobile magnetometer. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2017.

Khuong An Nguyen, Raja Naeem Akram, Konstantinos Markantonakis, Zhiyuan Luo, and Chris Watkins. Location Tracking Using Smartphone Accelerometer and Magnetometer Traces. *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pages 1–9, 2019. doi: 10.1145/3339252.3340518.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Müller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *arXiv*, 1 2012. doi: 10.48550/arXiv.1201.0490.

Lawrence R Rabiner and Ronald W Schafer. Introduction to Digital Speech Processing. *Foundations and Trends® in Signal Processing*, 1(1–2):1–194, 1 2007. ISSN 1932-8346. doi: 10.1561/2000000001.

Matthew Shaer. The Long, Lonely Road of Chelsea Manning - The New York Times, 6 2017. URL https://www.nytimes.com/2017/06/12/magazine/the-long-lonely-road-of-chelsea-manning.html.

Babins Shrestha, Di Ma, Yan Zhu, Haoyu Li, and Nitesh Saxena. Tap-wave-rub: Lightweight human interaction approach to curb emerging smartphone malware. *IEEE Transactions on Information Forensics and Security*, 10(11):2270–2283, Nov 2015. ISSN 1556-6021. doi: 10.1109/TIFS.2015.2436364.

Kevin Sun. Google Play Apps Drop Anubis, Use Motion-based Evasion, 1 2019. URL https://www.trendmicro.com/en_us/research/19/a/google-play-apps-drop-anubis-banking-malware-use-motion-based-evasion-tactics.html. Example of malware authors using an accelerometer to detect Googles bouncer and get their app onto the legitimate Google play store. The apps were called BatterySaverMobi and Currency Convertor.

Vladimir Vovk. Conditional validity of inductive conformal predictors. In *Asian conference on machine learning*, pages 475–490. PMLR, 2012.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. Algorithmic learning in a random world. *Journal of Computer and System Sciences*, 71(3):364–380, 2005.

Jiexin Zhang, Alastair R. Beresford, and Ian Sheret. SENSORID: Sensor Calibration Fingerprinting for Smartphones. *2019 IEEE Symposium on Security and Privacy (SP)*, 00: 638–655, 2019. doi: 10.1109/sp.2019.00072.