Predicting Keystrokes using an Audio Side-Channel Attack and Machine Learning

Samuel Ellis

University of Nottingham

A MSc Computer Science Dissertation submitted to the University of Nottingham
in requirements of the degree of Master of Science in Computer Science.

Submitted 10 September 2021

Abstract

Audio side-channel attacks are increasingly becoming a security concern regarding 'keystroke snooping', in which an attack can utilise the emanation of a keystroke to predict a specific key (or contextual passage of keys) being pressed. This can potentially be used to gather a users' private data if keystroke audio is able to be discretely captured. In this paper, acoustic emanation and geometric features are shown to provide enough information to accurately classify keystroke emanations. A combination of *MFCC* and *TDoA* features are shown to provide superior classification results when compared to other input features. A novel attack is presented which utilises cross-prediction techniques on a stereo array of microphones to increase keystroke recognition accuracy. Cross-predictions increase singular character recovery of keystrokes by 7% when using a supervised Random Forest machine learning model. A Random Forest classifier is able to achieve up to 89% inter-dataset single-character recovery from a 40-key classification problem. User experiments are also conducted to show the model in real-world scenarios. In the experiments, up to 85% keystroke recovery from contextual arguments were achieved from a 26-key classification problem using a Random Forest classifier. Keystroke recovery can increase by as much as 15% when utilsiing cross-prediction methods on contextual sentences. Contextual arguments were best predicted when using a user-created database of keystroke emanations. It is shown in this paper that different users emit distinct sonic fingerprints when typing on the same keyboard. Provided that a database of labelled keystrokes can be collected from a user, a supervised attack remains feasible in real-world scenarios.

Acknowledgements

I would like to give acknowledgements to those who have helped me throughout the completion of this project. First and foremost I am extremely grateful to my supervisor, Dr Jamie Twycross, for his invaluable guidance and support throughout the project. I would like to thank Rachel Lambourne for her unfaltering confidence in me. Finally, I would like to thank my family and friends for their patience and encouragement throughout the project.

1 Introduction

1.1 Background

Many computer tasks rely on the entry of data, such as logging into a user account, or composing a new email message. This data input usually depends on a computer keyboard to produce a desired word or key command depending on the use case. The mechanical nature of a keyboard means that keystrokes generated by a user produce an acoustic emanation that can be used to distinguish keys on a keyboard. Whilst the majority of these emanations may seen imperceivable to the human ear, the analysis of frequency content from different keys indicate that microscopic tonal differences exist between them. Utilising this information enables an audio side-channel attack using a keyboard and sound capturing device - an attacker can target the audio signature emitted from a keyboard to reconstruct a users' data input. This leads to serious privacy concerns if an attacker is able to discretely capture an audio recording of a users keystroke actions.

Previous research considering audio side-channel attacks on keyboards involve the exploitation of the acoustic emanation of keystrokes, as well as triangulation techniques in which information relating to the geometric position of a key can be determined (Asonov & Agrawal, 2004; Zhu et al., 2014). Contextual information using linguistic models have also been utilised to improve recognition rates (Zhuang et al., 2009). These methods have been proven to work with accuracies detailed in literature as high as 97.6% using a single mobile phone to collect a users data (Liu et al., 2015), and keystrokes have been shown to be deciphered using *Voice over IP (VoIP)* software such as Skype with accuracies reported to be as high as 91.7% (Compagno et al., 2017).

In this paper, the procedures taken to record keystrokes, fragment keystrokes into their respective classes, and develop a machine learning model to classify keystrokes from their respective features are undertaken. Figure 1 generalises the steps taken in this paper to classify keystrokes. The context of this paper within the abundance of available study into keyboard acoustic emanations is first

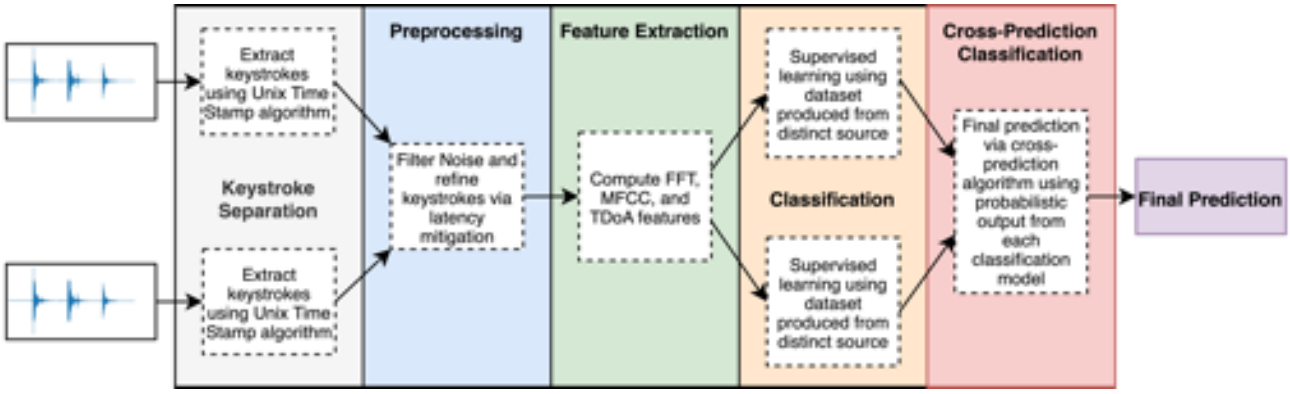


Figure 1: *An overview of the processes used to recover and predict keystrokes.*

discussed (Section 2). A keystroke dataset is then collected using an array of microphones (Section 3). Relevant features are extracted from this database (Sections 4 and 5) for use in a supervised classification algorithm which utilises cross predictions between microphone datasets to increase keystroke recognition rate (Section 6). Different users' experimental results are then analysed (Section 7). Lastly, conclusions from the whole report are discussed, alongside future research and mitigation strategies (Section 8).

1.2 Research Contributions

Important contributions from this report are highlighted below.

- A common pipeline in the data processing of keystrokes include the creation of multiple datasets for different ambient environments - for example, Zhuang et al., 2009 created alternative datasets for quiet and noisy environments. In contrast to this approach, state-of-the-art noise reduction techniques (using the third party software 'iZotope RX') will be used to reduce the noise within keystroke training and testing samples. A thorough search of related literature yielded a single paper from Hiu and Fiona, 2006 which concerned noise reduction. However, their user testing was limited and they did not conceptualise acoustic emanation features using their dataset.
- Instead of relying on one model for keystroke classification, the stereo array of microphones used in this research will each independently output probabilities for keystroke classification. Each probability matrix will be cross correlated to decide a final keystroke classification. From user tests, this can improve keystroke recognition rates by as much as 15% using a Random Forest model. This design choice has not been conceptualised by previous research regarding keystroke acoustic emanations.

2 Related Work

There are an abundance of research studies accessible involving the recovery of keystrokes from side-channel attacks. These related literature usually explore three alternative attacks - acoustic emanation/cryptanalysis approaches, geometric approaches, and linguistic approaches. Each technique has distinct advantages, which if combined can be used to exploit a users personal information. From the related literature, it is shown that acoustic emanation and geometric strategies in isolation have an approximate recovery rate of 70% (Asonov & Agrawal, 2004; Backes et al., 2010) and 72% (Zhu et al., 2014) respectively. Combinations of these methods provide even greater recovery rates as shown by Liu et al., 2015. The following sections detail the different approaches used to form acoustic emanation attacks in previous research.

2.1 Acoustic Emanation Approaches

Acoustic cryptanalysis attacks aim to target supposedly secure information which escapes as an indirect audio channel. These attacks can occur in many forms, such as the analysis of computer noise such as 'coil-whine', which is generated by capacitors. Eavesdropping on electromechanical components such as a computer keyboard can be completed using acoustic emanation attacks with high success (Genkin & Tromer, 2014).

The underlying principle of this attack method is that the emanation of a keystroke can be used to distinguish a specific key being pressed - the differences in acoustic emanation may be indistinguishable to the human ear, but feature analysis from a machine learning model can be used to identify unique keystrokes. The following sections describe two research studies using acoustic emanation approaches from Asonov and Agrawal, 2004 and Berger et al., 2006.

2.1.1 Asonov and Agrawal, 2004

The research by Asonov and Agrawal, 2004 is (reportedly) the earliest attack facilitating the use of keystroke acoustic emanations. Despite the age of this paper, the research details the vulnerabilities of keyboard acoustic emanation attacks on computer security by developing a supervised neural network to distinguish keystrokes. Their approach consisted of recording and training 100 keystroke samples of each key using a simple PC microphone at a recording distance of 0.5 meters, and a parabolic microphone for listening distances of up to 15 meters. Key findings from the paper show that detection accuracy decreases when applying the trained neural network on different keyboards and users.

The work presented by Asonov and Agrawal, 2004 provides general guidelines and the steps and processes used to create and train a dataset using acoustic emanation strategies. They provide a clear scope of whether the other side-channel techniques presented in Sections 2.2 and 2.3 must facilitate acoustic emanation strategies to provide more accurate results.

This research provides a concise overview of the exploration of acoustic emanations from computer keyboards, but still provides further directions of research, namely the exploration of environmental factors in skewing results using acoustic cryptanalysis methods - for example, the effect of ambient noise on the accuracy of a machine learning model, such as that in an office/workplace environment.

The approaches used in Asonov and Agrawal, 2004 can be seen as limited when considering a real world model:

- Their system does not take into account any macroscopic features of a keyboard, such the 'space', 'delete', and other special keys. These keys provide crucial information which can help when building a linguistic approach (as shown in research in Section 2.3).
- The accuracy rate of their model drops significantly when performing on another keyboard, even of the same model, due to the context-based 'fingerprint' required when performing supervised emanation feature extraction.
- Their approach only considered a mechanical keyboard, which emanates a more audible click when compared to modern PC keyboards which have less key travel and a less-distinguishable emanation.

2.1.2 Berger et al., 2006

Research conducted by Berger et al., 2006 combines signal processing and efficient algorithms to successfully classify 73% of single words of 7-13 characters from the recordings of keystrokes. Their research discovered that keystrokes which are in a close geometric space produce similar emanations to one another than keys that are further apart. Keystroke features are extracted from both the press and release peaks of a keystroke using *Fast Fourier Transform (FFT)* coefficients. Though the results from Berger et al., 2006 are effective when uncovering keystrokes from acoustic emanations, some limitations exist in their approach:

- In the same respect as Asonov and Agrawal, 2004, the approaches used by Berger et al., 2006 only use mechanical keyboards which produce a prominent acoustic emanation from each keystroke.
- The audio features extracted from keystrokes in their research were found to be inferior to other audio features analysed in this report (detailed in Section **??** *MFCC* audio features were found to perform better than the *FFT coeffients* used in their research.

2.2 Geometry-Based Approaches

Geometry-based approaches involve the use of more than one microphone to capture a keystroke signal - the signals are correlated to geometrically identify a keys physical position, as opposed to its given sound signature as described in Section 2.1.

This approach is interesting when considering the fact that a geometry-based method is context-free - it is not bound to a 'fingerprint' scheme like methods such as acoustic emanation, as it simply uses geometric attributes. Therefore, geometry-based approaches have distinct advantages when compared to acoustic emanation approaches (Zhu et al., 2014):

- An unsupervised learning scheme means this method does not have a definitive 'fingerprint' - a model trained with one keyboard can apply the same training dataset to another keyboard, whereas with a supervised fingerprint method (such as acoustic emanation) the training data is unlikely to accurately predict keystrokes from another keyboard (Asonov & Agrawal, 2004).
- Geometric methods are less susceptible to 'noise' interference, such as testing in a noisy office environment. Unlike an acoustic emanation approach, a 'fingerprint' comparison is not required, meaning it is not dependent on the sound signature of a captured signal.

The following sections describe two research studies using geometric approaches from Zhu et al., 2014 and Liu et al., 2015.

2.2.1 Zhu et al., 2014

Research from Zhu et al., 2014 involves the estimation of the physical position of a keystroke using a *Time Difference of Arrival (TDoA)* method. At least two stereo-enabled mobile phones (four microphones) are used in their approach to capture each keystroke, and each signal is cross-correlated to pinpoint a key position on a keyboard. Their method accurately recovers more than 72.2% of keystrokes in experimental situations such as a library, office, and meeting room.

This research provided a new alternative to a context-based attack, and is the first research to show the feasibility of keystroke recovery by a geometry-based method. However, there are limitations in their approach:

- The system designed in Zhu et al., 2014 requires a set of three stereo-enabled mobile phones to geometrically reconstruct a keyboard with their perceived accuracies. Using three collaborating phones makes this attack less feasible in real-world scenarios.
- Numerical keys (0-9) are not considered in their study, which may decrease the predicted accuracy of their chosen algorithm as this will account for approximate 36% increase in class distributions.
- Although the system proposed by Zhu et al., 2014 is context-free, it requires around 370 keystrokes from a target user to monotonically increase the accuracy from 18.1% to 72.2%.

The work presented by Zhu et al., 2014 provides useful information on pure geometric methods to determine a keystroke being hit, and their conclusion details interesting further research which will be recognised in this project - the combination of both context-based (emanation) and context-free (geometric) methods to further increase the success rate of key recognition.

2.2.2 Liu et al., 2015

Research conducted by Liu et al., 2015 combines the use of *TDoA* measurements alongside acoustic emanation features to cluster keystrokes into their respective classes. As opposed to research by Zhu et al., 2014, this method only uses one non-specialist piece of equipment such as a mobile phone to extract relevant information, and obtains a combinatorial accuracy of over 90%. This indicates that the combination of different methods can be used to identify keystrokes with significant accuracy.

Liu et al., 2015 also determine factors which affect the accuracy of keystroke detection using geometric methods - namely the sampling rate of the recorded source, and the distance between each microphone in the stereo system. This paper will be useful during the evaluation of each detection method. The research conducted by Liu et al., 2015 is the only single-device technique which enables acoustic snooping of keystrokes with great accuracy. However, this research does also not consider numerical keys, which could decrease the predicted detection accuracy, or special keys such as the space bar whose macroscopic qualities could significantly impact the accuracy of *TDoA* detection.

2.3 Linguistic-Based Approaches

Linguistic-based approaches involve the extraction of information from underlying content within the attack medium. Inaccuracies may still be present when determining keystrokes using emanation and geometric methods, with both methods (in isolation) having a detection accuracy of approximately 70% (Asonov & Agrawal, 2004; Backes et al., 2010; Zhu et al., 2014).

Linguistic methods aim to recover contextual information when inaccuracies are present in previous detection methods. Consequently, a linguistic approach can act as a final buffer to increase word recognition accuracy, if it is assumed that contextual information can be recovered.

Due to the time constraints of this project, it was not possible to develop a linguistics model for use in detecting keystrokes. However, research conducted by Backes et al., 2010 and Zhuang et al., 2009 provide useful insight into linguistic-based approaches for detecting keystrokes, and the provided literature is beneficial when considering future work for this project.

2.3.1 Backes et al., 2010

Research from Backes et al., 2010 presents an interesting combination of acoustic emanation and contextual linguistic analysis, but alternatively using this side-channel attack on printers. Their research details that using linguistic analysis alongside emanation methods leads to an increase in keystroke detection accuracy of approximately 20%. A combination of machine learning and audio processing techniques (such as speech recognition techniques and spectrum features) are presented in their research, which provides insight into the combination of attack approaches, albeit to a different problem.

One notable insight from Backes et al., 2010 are their use of speech recognition techniques, which could be implemented in the detection of keystrokes through the use of a *Hidden Markov Model (HMM)*. Backes et al., 2010 use this technique to rule out incorrect word combinations, at the expense of the assumption that the source target exhibits contextual properties.

3 Creating A Database

The first stage of developing a keyboard acoustic emanation attack is to collect well-defined keystroke signals which will be used for feature extraction when developing a machine learning model. The following sections describe the methods taken in this project to record, extract, and process keystrokes. The methods detailed in this section were used to create an inital training database of approximately 30 keystrokes per key class. alphabetical *a-z*, numerical *0-9* and special characters *backspace, enter, esc, and space bar* are considered in this database. This results in a 40 class dataset of approximately 1200 keystroke signals per audio source.

3.1 Recording Keystroke Audio

All parent files for keystroke audio were recorded using Python code. In this project, two external microphones were used to capture keystroke audio - a *Shure SM7B* and *Sonarworks XREF 20*. Audio was also simultaneously captured using a *16-inch MacBook Pro* internal microphone. Specifications of both external microphones are detailed in Shure, 2015 and Sonarworks, 2018 respectively. Both audio streams are recorded using the same clock source to compensate for clock drift influences on audio streams. The array of external microphones will be used to capture detailed acoustic emanations of each keystroke, as well as to extract time difference of arrival (*TDoA*) features detailed in Section 4. The latter hardware serves as an example for discrete invasion of a target's keystroke audio in real world scenarios, for example, an attack in a library setting where the attacker can discretely record from their laptop microphone while still being in close proximity to the target user. The primary keyboard under investigation in this project is the Apple A1644 keyboard [1] - a popular wireless keyboard commonly found in households and public computer facilities alike.

3.2 Extracting Keystrokes

As shown from Figure 2, a typical keystroke has three detectable peaks - a touch peak, hit peak, and release peak. In reality, the touch and hit peaks occur within milliseconds of one another, and the presence of ambient noise in the audio recording can make the touch and hit peaks even less pronounced. Therefore, a keystroke will be assumed to have two distinct peaks - a 'press peak' assigned to the touch and hit peaks, and a release peak.

It is shown in Figure 2 that a typical keystroke lasts between 100 and 150 milliseconds. This is concurrent with previous research regarding the capturing of keystroke emanations (Asonov & Agrawal, 2004; Liu et al., 2015). To create a dataset of individual keystrokes, all keystrokes are first captured and stored in a main audio file. Keystrokes must be separated from this file using their respective press and release peaks. The short elapsed time between two peaks of the same keystroke (as well as between adjacent keystrokes) mean that the detection and separation algorithms should suffice mm-level precision so data can be used to accurately train a final detection model.

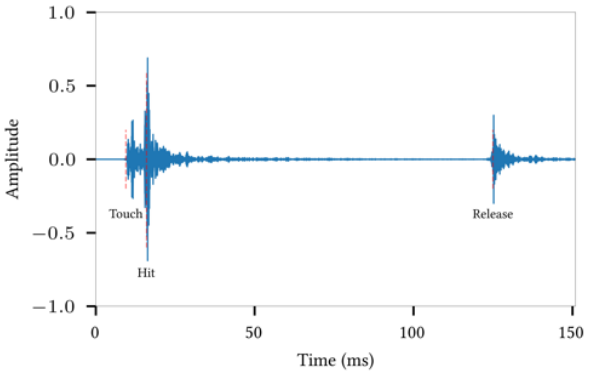


Figure 2: *The basic anatomy of a single keystroke signal showing a touch peak, hit peak, and release peak. Noise reduction has been performed on this keystroke signal to produce a model signal.*

The method opted for when collecting training data at this level of precision is through the use of Unix Time Stamps (*UTS*), which are used to determine the press and release peak of each keystroke relative to the start of the main audio file. This algorithm runs concurrently alongside the code used to capture keystroke audio (Section 3.1).

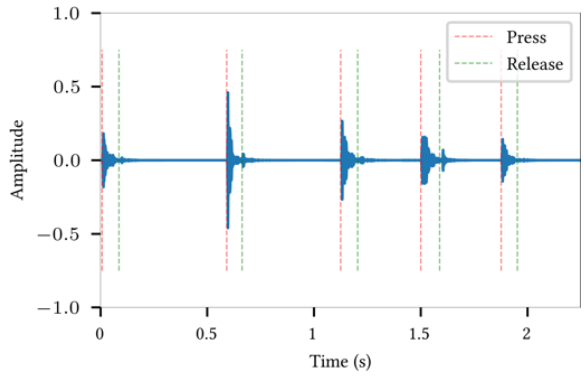


Figure 3: *The waveform of the keystroke sequence 'hello' after noise reduction. Unix time stamps have been used to determine each press and release peak in the audio-time series. Each event is used to isolate consecutive keystrokes.*

[1] In this paper, the Apple A1644 keyboard is referred to as *the keyboard*, unless explicitly stated otherwise.

When an audio recording is started, a *UTS* is simultaneously logged to determine the start of the audio region. When a key is pressed or released by the user, a respective *UTS* is logged for each event. The initial *UTS* is then subtracted from the *UTS* at the keystroke event, determining the time of the event relative to the beginning of the main file. All *UTS* events are appended to a JSON file with the same format shown in Section A, Listing 1. Figure 3 visualises *UTS* events on a sample audio file. Note that the algorithm displays inevitable error (as shown in the fourth keystroke slightly misinterpreting the evident release peak), potentially due to fluctuating software and hardware demands throughout the time of recording. Although the methods described in Section 3.3 could be used as an alternative algorithm to determine keystrokes, it is found that non-keystroke transient data and noise heavily influences the spectral flux envelope of keystroke signals - this method provides a more robust solution when considering external influences such as a ambient environment or computer fan noise.

3.2.1 Minimum Consecutive Keystroke Time

Denoting the captured Unix press and release times of a keystroke k as t_{kp} and t_{kr} respectively, to ensure the keystroke separation algorithm initially recovers the whole keystroke signal, 10 milliseconds before t_{kp} and 50 milliseconds after t_{kr} are separated into individual acoustic signals. Due to this separation method, it is assumed that neighbouring keystrokes from a user are not pressed within $(t_{kr} - t_{kp}) + 10\text{ms} + 50\text{ms} = 100\text{ms} + 10\text{ms} + 50\text{ms} = 160\text{ms}$ of one another to allow clear parsing of keystroke signals.

3.3 Audio Latency Mitigation

Section 3.1 details that three microphones will be used to capture the acoustic emanation of keystrokes. Two source devices will be used to record keystroke audio - the internal MacBook audio engine and an external multichannel audio interface. Although the source devices are recorded concurrently, a small latency still exists between both streams. Although the latency is a mm-level discrepancy, the average duration of a single keystroke is within the range of 100-200 milliseconds. This latency is to be mitigated before keystroke separation to ensure full keystroke signals are captured.

The transient nature of a general keystroke signal allows keystrokes to be located in an audio file due to their sharp increase in onset strength. This method assumes that background ambience is not able to significantly mask transient signals from a keystroke. In this paper, the *spectral flux* (a measure of how quickly the power spectrum of a signal changes) of a keystroke signal is determined to establish the exact moment a press peak is instantiated in time. The spectral flux of a keystroke signal is calculated using the `onset_strength` method provided by McFee et al., 2015. Figure 4 shows the onset envelope of a phrase of keystrokes. To determine the latency between two signals, the vertical lines shown by 4 are cross-compared for both source devices. The mean latency can be computed using Equation 1:

$$\Delta L = \frac{1}{n}\sum_{n=1}^{n} t_{1n} - t_{2n} \tag{1}$$

Where ΔL is the mean latency between both audio streams, n is the number of keystrokes in the audio file, and t_{1n} and t_{2n} are the press peak onset times in the first and second main files respectively.

3.4 Refining Individual Keystrokes

Individual keystroke signals still inherit a degree of error from the Unix time algorithm described in Section 3.2. For feature extraction to be reliable and standardised across all keystroke signals, the press peak of all signals must coincide within the first ten milliseconds of their audio file. This is achieved using the spectral flux feature of a keystroke, the same method used to mitigate latency in alternate audio stream (see Section 3.3). Instead of determining the spectral flux from the main audio file, the spectral flux of individual keystroke signals are determined. The developed algorithm determines a keystroke signals' onset time similar to the algorithm shown by Figure 4.

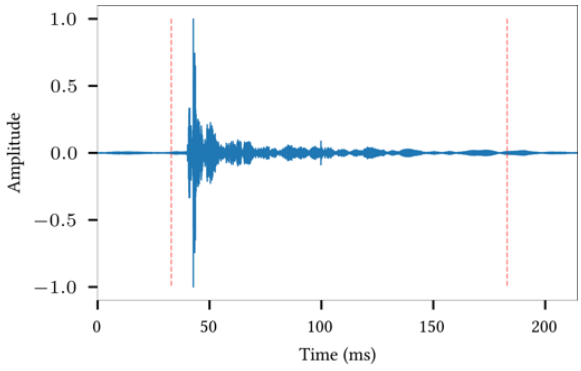


Figure 5: *A keystroke before refinement showing after spectral flux analysis. Audio data coinciding within the red vertical lines indicates a refined keystroke signal which will be used for feature extraction.*

By calculating the onset time of a single keystroke signal using the method visualised in Figure 4, the

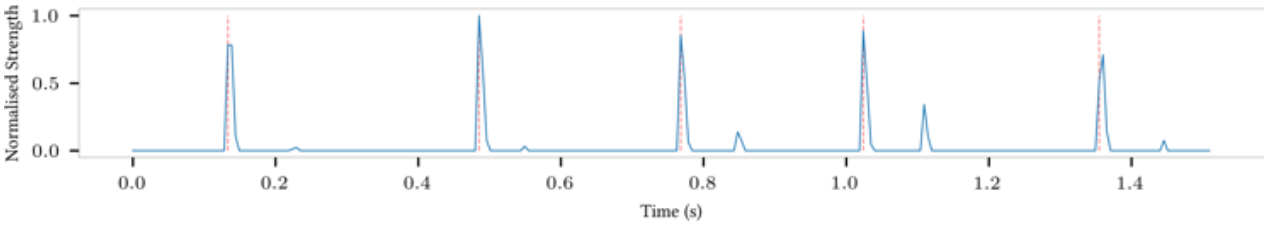


Figure 4: *The spectral flux onset envelope of a keystroke passage consisting of five keystrokes. Note the relative strength between press and release peaks of a corresponding keystroke, which shows a press peaks' robustness when refining keystrokes. The red lines indicate what the algorithm interprets as a press peak onset.*

exact time of a signals' press peak can be identified with minimal noise interference due to the removal of unwanted artefacts in the main audio file using the Unix time algorithm. Figure 5 shows the keystroke before and after refinement. To preserve geometric features for extraction in Section 4, timing information between respective signals of the same keystroke are unchanged, hence there is some fluctuation around the exact start time of each signal. Subsequently, ten milliseconds of audio is taken before determined press peak to ensure timing information is captured completely. A succeeding 150 milliseconds of audio is captured after the press peak to capture the whole keystroke signal.

3.5 Noise Reduction

The comparatively weak acoustic intensity of a keystroke relative to room ambience means that microphone pre-amplifiers must be operated at a greater input gain. This inevitably increases the noise floor and pre-amplification noise within the recording environment, which imparts more noise into a keystroke recording. To ensure each machine learning algorithm extracts information solely from the keystrokes, noise reduction is performed on all keystroke signals. Due to the time constraints in this project, noise reduction on the dataset will be conducted using commercial software. Noise reduction on individual keystrokes is done using the *Spectral De-noise* module in the *iZotope RX* software suite. Figures 6a and 6b show the keystroke signal before and after rendering respectively - it is shown that the majority of unwanted noise is removed from the signal, which will result in precisely-defined features upon extraction.

4 Acoustic Feature Extraction

The following sections detail feature analysis provided from the acoustic emanation of keystroke signals.

4.1 Frequency Spectrum Analysis

Though comprehensive keystroke signals have been captured using the methods defined in Chapter 3, the frequency information emitted through the duration of a keystroke signal may be unevenly distributed. Figure 7 shows the frequency information given by an example keystroke signal, where delta is the time elapsed between the press and release peaks of a keystroke.

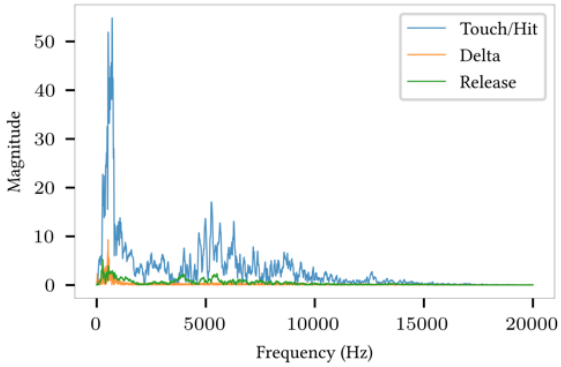
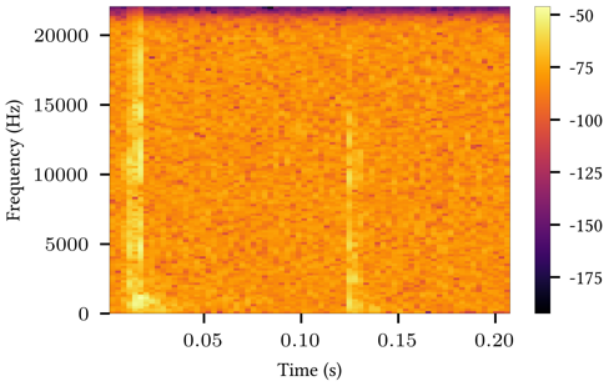


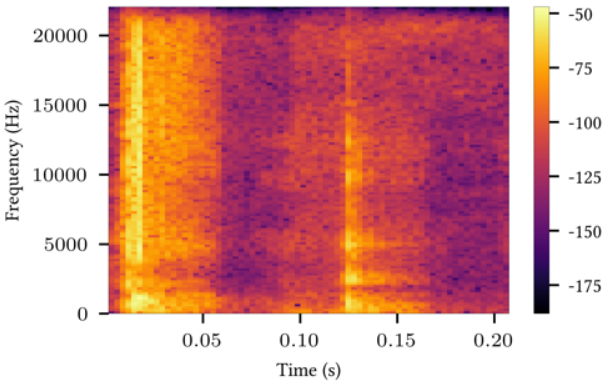
Figure 7: *The frequency spectrum of the touch/hit peak and release peak from the space bar signal. Delta signifies frequency information residing between both peaks.*

By splitting the signal into three separate components for analysis, it is shown that the majority of useful frequency information comes from the press peak of a keystroke signal. A keystroke release peak could also be used to add subsidiary information to a classification scheme, though this will not be considered in this paper. The delta information seems to hold minimal useful frequency information when classifying a signal. Subsequently, press peak frequency information will solely be used for acoustic emanation analysis. While significant information is present in the release peak of a signal, project time constraints ruled out the possibility of dissecting release peaks from keystroke signals.

The frequency spectrum in Figure 7 shows a fre-



(a) *Before noise reduction*



(b) *After noise reduction*

Figure 6: *A keystroke signal before and after noise reduction using iZotope RX Audio Editor.*

quency range of 20-20000 Hz as this is a typical range where important acoustic information resides, as well as the frequency range of the Sonarworks XREF 20 microphone used to record keystroke samples (Sonarworks, 2018). However, it can be seen from Figure 7 that the majority of useful frequency information lies in the 400-14000 Hz range for a push peak of a keystroke signal, with frequency bias in the 400-7000 Hz range. This frequency range agrees with previous research from Asonov and Agrawal, 2004 and Zhuang et al., 2009. Because of this, a range of 400-14000 will be considered for emanation feature extraction to reduce artefacts in classification analysis.

4.2 Classification Feasibility

This section briefly details the feasibility of using frequency information emitted from a keystroke to classify keys into their respective classes. Figure 8 shows the mean frequency spectrum of 30 keystroke signals from the key classes 'p' and 'l'.

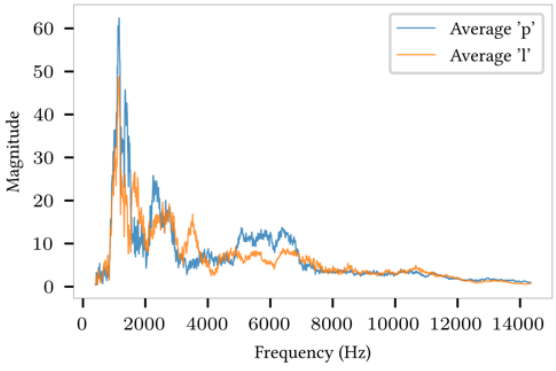


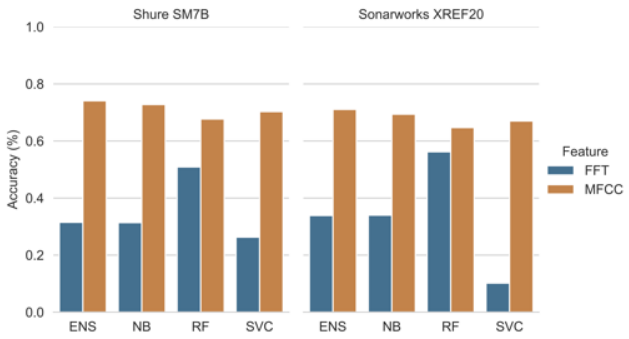
Figure 8: *The average frequency spectrum of the two key classes 'p' and 'l'. 30 keys were analysed per key class to produce an average curve.*

It is shown from Figure 8 that the average frequency information in separate key classes are distinguishable. Class 'p' keystrokes are shown to be identifiable by sharp frequencies in the 2000 and 3800 Hz ranges, and class 'l' keystrokes are shown to have characteristic peaks in the 1800 and 3600 Hz area. It is noted that valuable information for both classes resides in the 400-7000 Hz range, which coincides within the telephonic audio range provided by most *VoIP* commercial software (Compagno et al., 2017)

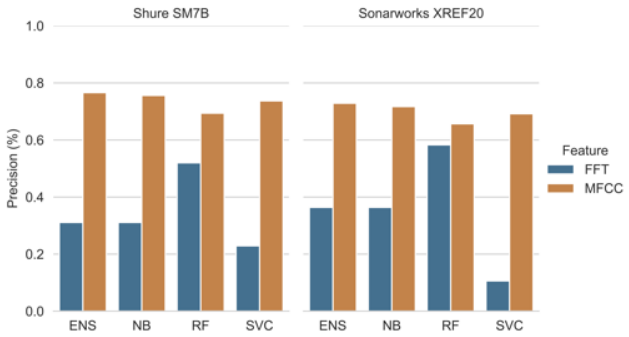
4.3 Acoustic Classification Techniques

To classify keystrokes using acoustic emanations, *MFCCs* and *FFT* coefficients were extracted from each keystroke signal. *FFT* features were calculated using a window size of 10 milliseconds (960 samples at 96 kHz sampling rate) and a hop length of a quarter of the window size. The same window size and hop length were used to extract *MFCC* features. In this case, the first 13 *MFCCs* were used as features from each window. *MFCC* and *FFT* features are compared against four different classification models in Figure 9. Only frequency data in the 400-14000 Hz is considered as useful, hence frequency information residing outside this range is ignored upon feature extraction.

From Figure 9, it is shown that *MFCCs* provide better classification rates when compared to *FFT* coefficients, regardless of classification model used. The highest keystroke recognition rate achieved by *FFT* features when using data from the Shure SM7B dataset was 54% (RF), whereas the highest recognition rate achieved using *MFCCs* as a feature was 76% (ENS). *FFT* features has a lower keystroke recognition accuracy when compared to their respective *MFCC* features for all the classifiers that



(a) *Accuracy metrics.*



(b) *Precision metrics.*

Figure 9: *Single character classification metrics using FFT coefficients and MFCCs as input features. Four different classification algorithms are analysed - Random Forest (RF), Naive Bayes (NB), Support Vector Classifier (SVC), and an Ensemble (ENS) classifier constructed from the former classifiers. The classification metrics shown are achieved using a sampling rate of 96 kHz.*

were investigated. This concurs that *MFCCs* are a superior feature to *FFT* coefficients when classifying keystrokes, which also agrees with relevant research regarding the classification of keystroke signals (Zhuang et al., 2009; Anand & Saxena, 2018).

5 Geometric Feature Extraction

The following sections describe theory regarding the extraction of Time Difference of Arrival (*TDoA*) features from keystroke signals. *TDoA* is defined as the elapsed time of a signal between two stationary microphones - the difference in time of arrival between the two microphones indicates a geometrical location for the source signal. The keyboard and microphones are positioned according to Figure 10.

5.1 Theoretical Maximum Lag

Due to the high sample rate used in recording all keystroke signals (96 kHz), the methods defined in Section 5.3 need to work efficiently for fast feature extraction. Therefore, it is important to set a theoretical maximum *TDoA* for all keystroke signals. Figure 10 outlines the mathematical model used for determining the maximum *TDoA*.

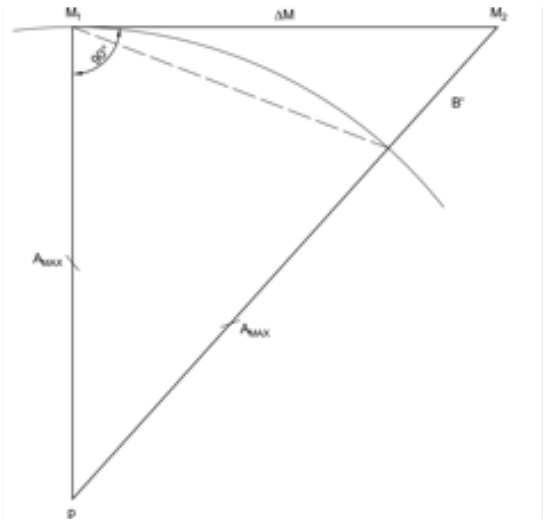


Figure 10: *A schematic used to determine the maximum time-difference of arrival of a keystroke signal.*

From Figure 10, both microphones M_1 and M_2 are placed in a fixed position and are separated by distance ΔM. The array of microphones are assumed to be parallel with the keyboard under investigation. Keystroke p represents the bottom leftmost corner on the keyboard. The distance between M_1 and p identifies A_{max} - the maximum distance a keystroke emanation has to travel to reach M_1. Assuming the lines M_1p and M_1M_2 are perpendicular with one another, simple trigonometry can be used to determine the maximum distance of arrival, B' (in metres):

$$B' = \left(A_{max}^2 + (\Delta M)^2\right)^{0.5} - A_{max} \tag{2}$$

The distance of arrival calculated in Equation 2 can be divided by the velocity of sound, v, and multiplied by the sampling rate of the signal, F_s to obtain the theoretical maximum *TDoA* in samples, τ. The finalised equation to determine τ is shown by Equation 3. Assuming each microphone will be placed at each end of the keyboard, Figure 11 shows the maximum theoretical *TDoA* for an Apple A1644 and Logitech K780 keyboard. From the schematic detailed in Figure 11, the maximum distance a keystroke has to travel (A_{max}) is 15 centimetres. This results in an upper-bound theoretical TDoA of 50 and 80 samples respectively when $F_s = 96000$.

$$\tau = \frac{B' \cdot F_s}{v} \tag{3}$$

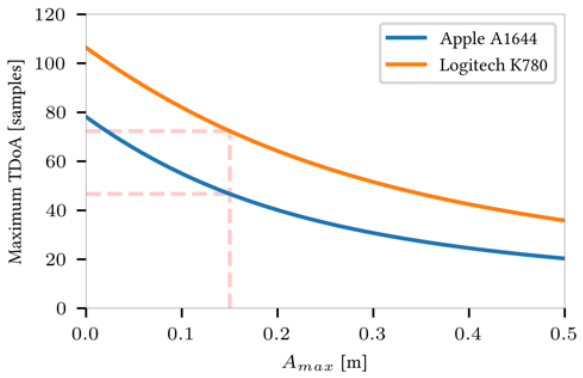


Figure 11: *A plot of the maximum distance a key has to travel (A_{max}) and its respective maximum theoretical TDoA for an Apple and Logitech keyboard.*

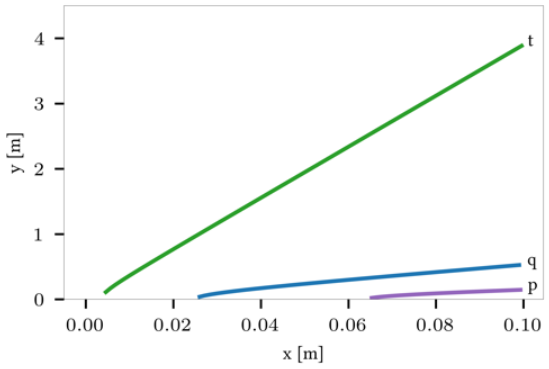


Figure 12: *A visualisation of Equation 4 for three geometrically-discrete keys on an Apple A1644 keyboard.*

5.2 Theoretical Angle of Arrival

From research conducted by Scola and Ortega, 2010, the angle of arrival of a signal can be determined from a two-source microphone array using its *TDoA*, assuming that microphones M_1 and M_2 are stationary reference points on the same plane and a microphones distance from the source is greater than five centimetres. Equation 4 from Scola and Ortega, 2010 can be used to determine the y-coordinate from a given x-coordinate of a keystroke signal. If both coordinates are known, the angle can hence be determined. Figure 12 shows Equation 4 plotted for different geometrically-placed keystroke signals. It is assumed that the lines plotted in Figure 12 are linear such that a gradient can be calculated from each line to describe an approximate y-coordinate from given x-coordinates. Substituting this value into Equation 5.2 provides the theoretical angle of arrival for a keystroke signal (Scola & Ortega, 2010).

$$y = \pm\left[\frac{B'^2}{4} - x_{M_1}^2 + x^2\left(\frac{4 \cdot x_{M_1}^2}{B'^2} - 1\right)\right] \tag{4}$$

$$\theta = \begin{cases} -\left(90 - \tan^{-1}\left(\frac{dy}{dx}\right)\right), & \text{if } \tau < 0, \\ 90 - \tan^{-1}\left(\frac{dy}{dx}\right), & \text{if } \tau \geq 0. \end{cases} \tag{5}$$

Where $^{dy}/_{dx}$ is the gradient of the respective line shown by Figure 12) of the keystroke under investigation. Table 1 shows the average angle of arrival calculated using Equation for 30 keystroke signals for the key classes 'qztvmp', versus their measured angle of arrival. This shows that whilst the predicted angle varies from the actual angle significantly for some keys, the angle of arrival can be used to determine an approximate keystroke location alongside *TDoA* features.

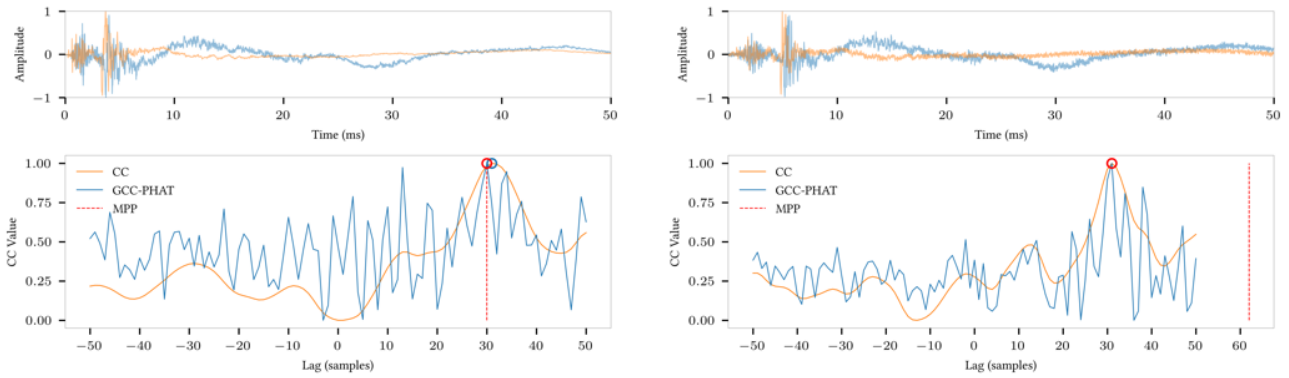
Table 1: *Mean theoretical angle of arrival (*deg*) from 30 'qztvmp' keystrokes in each respective class, versus each keys actual measured angle. PC% is the percentage change between actual and predicted angle of arrival.*

Key	Predicted θ	Actual θ	PC%
q	12.4	28.0	126
z	6.60	23.0	248
t	-2.20	9.00	-509
v	-3.40	5.00	-247
m	-11.6	-10	-13.8
p	-14.9	-32.0	-115

5.3 Determining TDoA from Keystroke Signals

This section details three methods which have been analysed in this paper to extract *TDoA* features from keystroke signals - *Maximum Peak Position*, *Cross Correlation*, and *Generalised Cross Correlation with Phase Transform*.

Maximum Peak Position (MPP). The generalised keystroke signal visualised in Figure 2 shows that a perfect keystroke exhibits a maximum amplitude near the hit peak of the signal. Assuming the maximum amplitude in two signals of the same



(a) *A clean (no ambient noise) signal.*

(b) *A noise-reduced (but still noisy) signal.*

Figure 13: *Two 'z' keystrokes analysed for TDoA feature extraction with respect to MPP (maximum peak position), CC, and GCC-PHAT techniques.*

keystroke exhibit this exact behaviour, the *TDoA* can be determined from the onset time of both peaks. This method assumed that a keystroke signal is not significantly influenced by external noise or internal microphone noise. However, in ambient environments the noise may significantly affect the relative amplitude of a signal.

Cross Correlation (CC and GCC-PHAT). Cross-correlation techniques aim to measure the similarity between two discrete audio-time series of the same signal, and return an estimated lag time (in samples) between both signals MATLAB, 2021. Two methods concerning cross-correlation techniques will be used in this report - *CC* and *GCC-PHAT*. The latter technique is commonly used to achieve robustness with respect to reverberant and noisy signals, so is best suited for determining the *TDoA* of a keystroke in ambient environments (Lee et al., 2020).

To determine the best *TDoA* calculation method, a keystroke sample and its respective reference signal were plotted alongside a lag-plot for each method above, using the theoretical maximum lag determined in Section 5.1 as an upper-bound. Figures 13a and 13b show each techniques detection ability with respect to a clean keystroke signal and a noisy keystroke signal.

From Figure 13, it is clear that the *MPP* technique is not robust with respect to ambient noise and varies significantly with noise, even after noise reduction. Figure 13b returns a *MPP TDoA* value of approximately 60 samples - around a 30 sample discrepancy from the rest of the techniques shown in Figure 13, which equates to a 10.72 centimetre misinterpretation if converted from samples to distance metrics. Given that each key on a keyboard key is approximately 2 centimetres wide, this method provides unsuitable metrics for *TDoA* classification.

A sharp peak closer to unison in Figure 13 indicates the probable *TDoA* measurement for the signal. Both the *CC* and *GCC-PHAT* techniques provide accurate metrics when determining the *TDoA* in both clean and noisy environments, with each returning a *TDoA* of approximately 30 samples. However, the *GCC-PHAT* algorithm exhibits frequent sharper peaks compared to a single broad peak exhibited by the *CC* algorithm, as is shown in Figure 13a. Though both algorithms obtain a similar *TDoA* metric, the *GCC-PHAT* algorithm is considered to be easily misinterpreted for clean keystroke signals due to the magnitude of sharper peaks shown in Figure 13a, whereas in both scenarios the *CC* algorithm exhibits a broader curve but within a suitable range of acceptance. This indicates that the *CC* algorithm is best suited for keystroke *TDoA* feature extraction.

As Figure 13 only visualises metrics for one keystroke it may misrepresent the quality of each detection algorithm, therefore Table 2 is used to provide generalised results for a range of key classes. Results from Table 2 are created using 30 keys from each key class, making them representative of the keystroke database created in Section 3. Metrics from Table 2 show that the *MPP* algorithm provides unsuitable *TDoA* features for all analysed key classes due to their high variance when compared to cross-correlation methods. Both the *CC* and *GCC-PHAT* algorithms extract keystrokes within a suitable range of acceptance for each key class. *GCC-PHAT* metrics usually exhibit a larger degree of variance, which is likely due to the design of this algorithm for reverberant signals, whereas the analysed keystroke database has undergone noise reduction as described in section 3.5. Therefore, to mitigate inaccuracies in *TDoA* measurements, the *CC* algorithm will be used for general feature extraction, and the *GCC-PHAT* algorithm will be used

in the case of a noisy keystroke signal.

Table 2: *Mean TDoA values and standard deviation statistic for MPP, CC, and GCC-PHAT cross-correlation algorithms. Tested on 30 keys with respect to 'qztvmp' key classes.*

	MPP		CC		GCC-PHAT	
	Mean	STD	Mean	STD	Mean	STD
q	61.5	62.1	41.9	0.573	39.7	6.59
z	50.9	206	31.1	0.562	31.0	2.04
t	-33.3	118	12.2	10.3	7.7	15.5
v	-59.1	129	10.2	0.559	9.60	0.955
m	-65.7	107	-9.60	10.3	-11.0	2.86
p	-67.2	115	-24.8	18.7	-30.3	1.61

6 Keystroke Classification

When applying a supervised learning method to the dataset, *MFCC* and *TDoA* [2] features were used to classify each keystroke - details on the extraction of each feature are presented in Sections 4 and 5 respectively.

40 key classes which consist of approximately 30 keystrokes per class are featurised and inputted into multiple classifiers for initial analysis. All input features are first standardised to ensure data is internally consistent. Classifiers were optimised using the `GridSearchCV` function provided by Pedregosa et al., 2011 to determine the optimum parameters for all classifiers. Two datasets (from the stereo-microphone setup detailed in Section 3) were split into a labelled training and testing set with a respective 80/20 split as described by the Pareto principle (Nisonger, 2008). For each classifier, Five cross-validations were then performed on the Random Forest (RF), Naive Bayes (NBC), Support Vector (SVC), and Ensemble (ENS) classifiers to determine the keystroke recognition rate of each algorithm, alongside precision, recall, and F1 metrics. The results from this process pipeline for all classifiers are shown in Figure 15.

6.1 Combining Emanation and Geometric Features

In this section, *MFCC* and *TDoA* features are combined upon classification to determine their combinatorial effects on keystroke detection accuracy. Figure 14 details the recovery rate (accuracy) and precision metric for keystroke signals using *MFCC*, *TDoA*, and the combination of both features.

It is observed from Figure 14 that the *TDoA* feature of a keystroke performs the worst as an individual feature. The best accuracy and its respective precision score by only *TDoA* features are 37.3% (ENS) and 32% (ENS) respectively, indicating a combinatorial approach is required between acoustic emanation and geometric features for geometric classification to be beneficial. From all of the observed classifiers, *TDoA* features are inadequately classified by the SVC algorithm with accuracy and precision metrics stated as low as 7.1% and 2.3% respectively. The low precision metric in this model indicates that the classifier returns significant false positives for the testing dataset, and the majority of keystroke signals are incorrectly classified.

From Figure 14 it is concluded that the combination of input features significantly increases performance for the RF, whereas for the Naive Bayes and Ensemble classifiers this has a negligible effect on accuracy and precision. As shown by Figure 14b, the combinatorial effect when classifying using the SVC algorithm decreases the precision in the XREF20 model, meaning this algorithm is not suitable for the keystroke classification task. Combinatorial features in RF models from both datasets result in an increase in keystroke accuracy from 68.6% to 82%, an approximate 13% increase in correct classifications on the test data. The RF combinatorial algorithm also boasts similar precision scores showing the key classifications have few false positives.

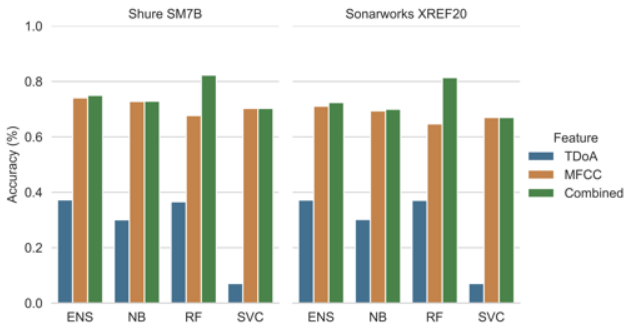
6.2 Choosing a Suitable Classifier

It is shown from 15 that the RF classifier performs significantly better keystroke classification when compared to ENS, SVC, and NBC classifiers, with the SVC classifier performing the worst for keystroke classification. Metrics for the RF classifier are as high as 82% accuracy and 84% precision considering the SM7B dataset in solidarity, compared to the ENS classifier which classifies with 68% accuracy and 70% precision - a 12% increase in keystroke recovery. All classifiers perform similarly with respect to microphone datasets in unison - as an example, the RF classifier predicts 82% and 79% accuracy metrics using the SM7B and XREF datasets respectively. This suggests that microphone features are independent when considering the accuracy of keystroke classification, provided that each model uses testing and training data collected from the same source microphone. As the RF algorithm showcases the best classification potential using train test data splits, this classifier will be used for the user classification tasks detailed in Section 7.

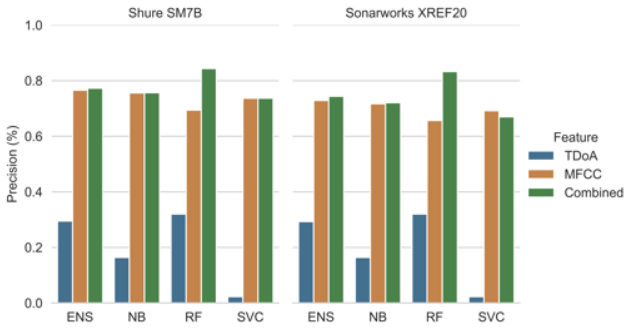
6.3 Cross Predicting Source Signals

Due to the stereo array of microphones used to record keystroke signals in Section 3, two Ran-

[2] *TDoA* features in this section refer to both the time-difference of arrival and angle of arrival, unless stated otherwise.



(a) *Accuracy metrics.*



(b) *Precision metrics.*

Figure 14: *Single character classification metrics using TDoA and MFCCs as input features. 'Combined' consists of creating a model that utilises both input features.*

dom Forest classification models can be used to predict each keystroke. Instead of each model predicting the class of a test keystroke signal in solidarity, they can be cross-examined to increase overall keystroke recovery. Instead of each classifier outputting a unary class, an array of probabilities are returned which details the likelihood of the example keystroke belonging an individual class as a percentage. This accomplished using functions provided by Pedregosa et al., 2011. Figure 16 visualises the probabilistic output of a sample '4' keystroke from the test data, using two unique Random Forest algorithms trained on both microphone databases. In this example, only numerical key classes are considered (*0-9*) to simplify visual analysis of cross predictions.

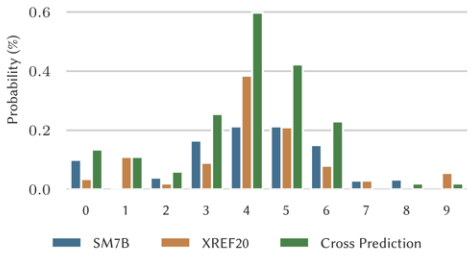


Figure 16: *A sample '4' keystroke being classified by two discrete Random Forest algorithms, using 0-9 keystroke training data from both the (Shure) SM7B and (Sonarworks) XREF20 datasets.*

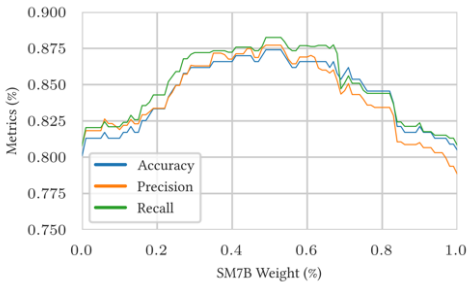


Figure 17: *The effect of changing the relative weightings for the SM7B and XREF20 datasets with cross-predictions, plotted against accuracy, precision, and recall metrics.*

It is shown from Figure 16 that each Random Forest algorithm can classify the '4' keystroke with reasonable probabilistic likelihood, as both algorithms show a respective 40% and 60% chance of classifying the keystroke correctly in unison. However, looking at the probabilities outlined by the '4' and '5' key classes using the SM7B dataset shows that the classifier trained on this dataset has a low precision, with each class outputting an approximate 20% probability for both classes.

Combining the probabilistic output from both algorithms mitigates the precision outliers highlighted from the SM7B dataset in this example. Assuming an equal weighting metric between each classifier, the summation (cross-prediction) of all values in each probability matrix classifies the key with an approximate 20% probability confidence interval with respect to the '5' class. This shows that even if one classifier can determine a keystroke with reasonable precision, a cross-predicted relationship between both classifiers can increase the precision for the final result.

It is important to note that Figure 16 also details the geometric relationship between acoustic emanations of different key classes. It was first hy-

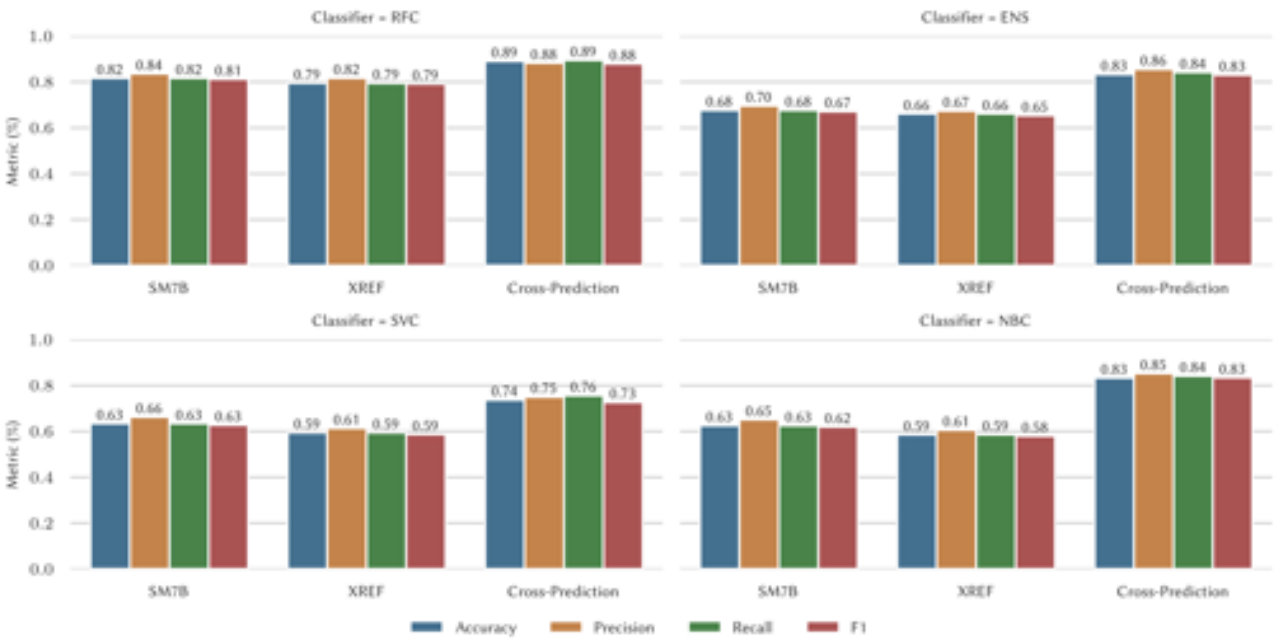


Figure 15: *Single character classification metrics using a combination of MFCC and TDoA input features. Four different classification algorithms are shown - Random Forest (RFC), Naive Bayes (NBC), Support Vector (SVC), and an Ensemble (ENS) composed from the former classifiers. Database classification in solidarity and using cross predictions are also shown for all classifiers. Metrics shown are achieved at a sampling rate of 96 kHz.*

pothesised by Berger et al., 2006 that the sound emanated by a keystroke correlates to its physical positioning on a keyboard. The bell shape distribution shown from Figure 16 agrees with this hypothesis - individual models predict higher probabilities for key classes with similar physical positioning as the actual key, and lower probabilities for key classes spaced further apart.

Figure 17 shows accuracy, precision, and recall metrics of overall keystroke recovery both databases versus different relative weights between the SM7B and XREF20 databases for cross-predictions. This data concludes that cross-predictions increase the recognition rate of keystrokes when the relative weights between both datasets lie in a range between 30% and 70%. This is confirmed using the train-test metrics visualised in Figure 15, where cross predictions are shown to increase keystroke detection accuracy by 7%, 16%, 9%, and 20% for RFC, ENS, SVC, and NBC classifiers respectively.

7 Experimental Results

The following section studies keystroke recovery rates from different users, using the keystroke training set used throughout Section 6, as well as user-created datasets. To collect samples from each user, the same setup is used as shown in Figure 10, with both the external microphones and keyboard studiously placed with the same relative geometry as detailed in Section 5. Because of this, studies solely on the differences in keystroke emanation between typists can be conducted.

To study the effects on typing styles from different users, alternate datasets were developed from each user, utilising three different typing approaches commonly referred to in keystroke emanation literature as Straw Man Typing, Hunt and Peck Typing, and Touch Typing (Halevi & Saxena, 2012). Each technique is briefly described below.

Straw Man Typing was first acknowledged by Asonov and Agrawal, 2004 and utilises a users index finger to continually type the same key a specified number of times. There is ample time between each successive keystrokes to ensure each keystroke can be captured cleanly. This technique is used to minimise differences in the acoustic fingerprint of keystrokes of the same class, as a similar metric of force will be instilled on each keystroke. This ensures a low degree of dynamics between adjacent key presses. This technique is utilised in Section 7.2.

Hunt and Peck Typing shares the same characteristics as straw man typing concerning the use of a singular finger to record keystrokes, except that consecutive keystrokes pressed by the user can

differ in class from one another. This induces an extra degree of dynamics into individual keystrokes, as the angle at which a users finger hits a key is influenced by previous keystrokes (Halevi & Saxena, 2012). Hunt and peck typing analysis is conducted in Section 7.3.

Touch Typing involves the use of multiple fingers to produce keystroke emanations. Multiple fingers may be used to strike the same key, and this typing scheme displays the largest degree of dynamics in a keystrokes respective acoustic emanation. This style of typing aims to replicate a users' actual typing mannerisms in perfect conditions. The only exemption between this typing form and a users authentic typing characteristics is the assumption that consecutive keystrokes do not overlap within a 130 milliseconds of one another, which was explained in Section 3.2.

7.1 User Typing Mannerisms

The two users who are under investigation can be qualitatively identified by their different typing mannerisms. The first subject (**User 1**) can be described as an average-speed typist (approximately 40 *WPM*) with heavy touches on keys which produces transient-enhanced acoustic emanations with respect to the press peak of each keystroke signal. This users' typing mannerisms are stylistically similar to that used when building the initial training dataset.

The second subject (**User 2**) can be described as a slower typist (approximately 32 *WPM*) who exhibits greater dynamics in their typing - transient emanations with respect to the press peak of a keystroke are less pronounced than User 1 and keystrokes display significant dynamic fluctuations with consecutive keystrokes. This users' typing mannerisms are stylistically different to that used to create the initial training dataset.

7.2 Straw Man Typing

For both users, a dataset was created using Straw Man Typing (*SMT*) mannerisms. The data collected in each dataset consisted of approximately 15 keystrokes of each respective class, recorded in the same environment and conditions as the main training database. The figure below shows different classifier metrics when using the training dataset from Section 6.2 and testing using the three databases created by each user. A Random Forest algorithm is used to classify keystroke. From the figure below, it is shown that the keystroke recovery rate for User 1 and User 2 are 51% and 57% respectively, with both datasets scoring precision metrics of 60% and 64%.

The metrics presented by Figure below are significantly lower than those detailed in Figure 15, which were quantified using intra-dataset train-test splits as opposed to inter-dataset predictions in this section. This hypothesises that specific fingerprints are created by distinct users dependent on their typing mannerisms. As the training dataset was created using only a single users typing, it has potential to heavily penalise classification for users who contrast from this typing behaviour.

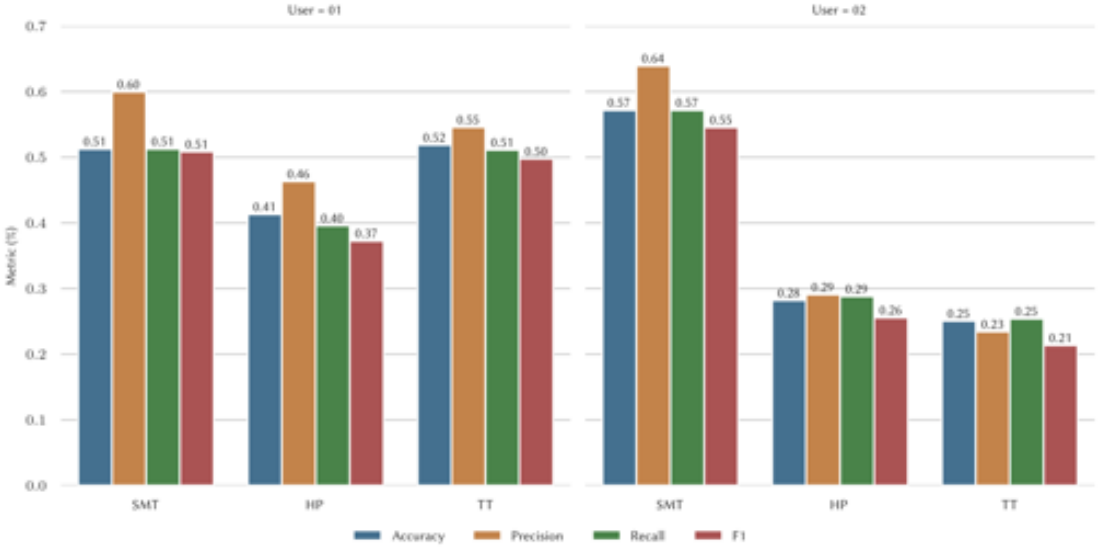


Figure 18: *Single character cross-predicted classification metrics from user created datasets using the Random Forest model determined in Section 6.2. The dataset from Section 3 is used to to predict each user database. The metrics shown are achieved at a sampling rate of 96 kHz.*

7.3 Hunt and Peck Typing

One dataset was also collected from each user concerning *Hunt and Peck Typing (HPT)* mannerisms. Once again, approximately 15 keystrokes from each class were collected per database, and ambient/environmental conditions conformed those when creating the main training dataset. To collect user keystroke data, users were asked to type 15 holoalphabetic sentences (a sentence where every letter of the alphabet is used at least once) into the algorithm, which approximates a dataset of 15 keystrokes per class. Numerical classes (*0-9*) and some special characters (*Esc, Backspace*, and *Enter*) were ignored in user datasets due to infrequent appearance of these keys in contextual sentences.

Figure 18 shows the predictive results for each *HPT* database per user, again using the training set from Section 3 to predict results. Metrics given by Figure 18 show that *HPT* mannerisms attain lower keystroke recovery rates than *SMT* databases for the same user, with accuracy scores of Users 1 and 2 detailed as 46% and 28% respectively. This is coherent with the theory described in Section 7.2 as *HPT* mannerisms induce dissimilarities in keystroke emanations of the same class.

User 2 exhibits a substantial drop in accuracy, precision, and recall metrics between *SMT* and *HPT* databases when compared to User 1. It was noted from the experiment that User 2 displayed greater dynamics in their typing, producing less-pronounced features when compared to User 1. This dynamic variance constitutes to the disparity in keystroke recovery rates concerning both users.

7.4 Touch Typing

Touch Typing *TT*) databases for each user were collected using the same strategies identified in Section 7.3. Metrics for each users' *TT* database are shown in Figure 18. User 1 exhibits a *TT* database accuracy of 52%, approximately a 9% increase from their respective *HP* classification accuracy. This may be due to the users' typing mannerisms as described in Section 7.1, in which they produce similar keystroke emanation patterns that were used to compose the training data set. The keystroke recovery accuracy is far lower upon testing the *TT* database from User 2, scoring 25% recovery rate with lower precision and recall metrics than their supplemental databases. Once again, the greater dynamic typing style exhibited by User 2 produced less pronounced features compared to the training dataset, hence the significant drop in keystroke recovery.

7.4.1 Combining Databases to Improve Keystroke Recovery

Figure 19 shows each users' cross predicted *TT* database metrics plotted with multiple dataset combinations that were used as training data for the Random Forest algorithms. Respectively, Figure 19 details that a combination of *SMT*, *HP*, and the external training data provided the best classification metrics for User 1 (59% accuracy). However, a combination of only *SMT* and *HP* databases provided the best metrics for User 2 (53% accuracy).

This reinforces the hypothesis that users generate a specific emanation fingerprint independent

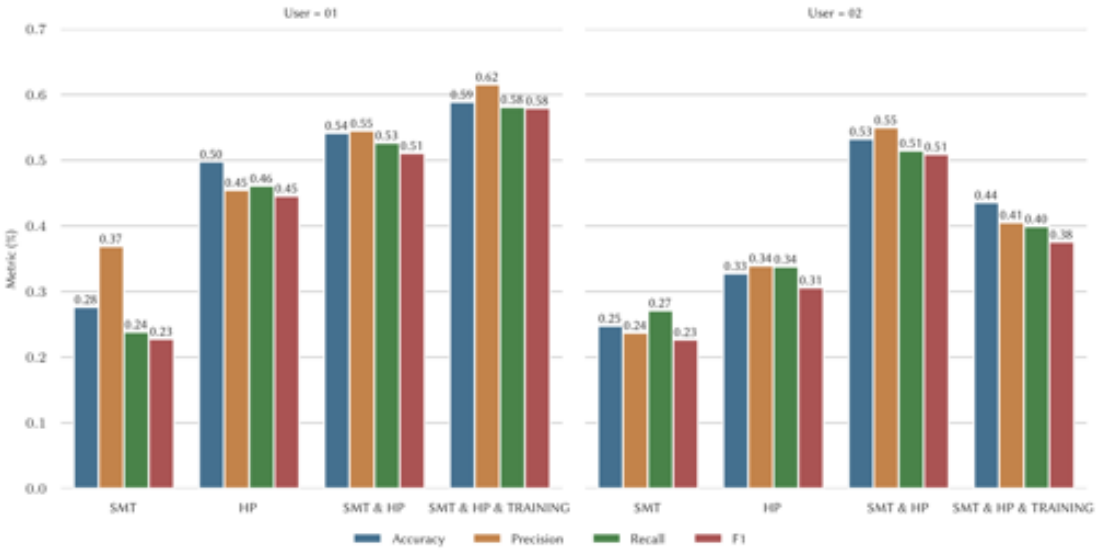


Figure 19: *Single character cross-predicted classification metrics from each users' TT database, compared against different combinations of user datasets. The metrics shown are achieved at a sampling rate of 96 kHz.*

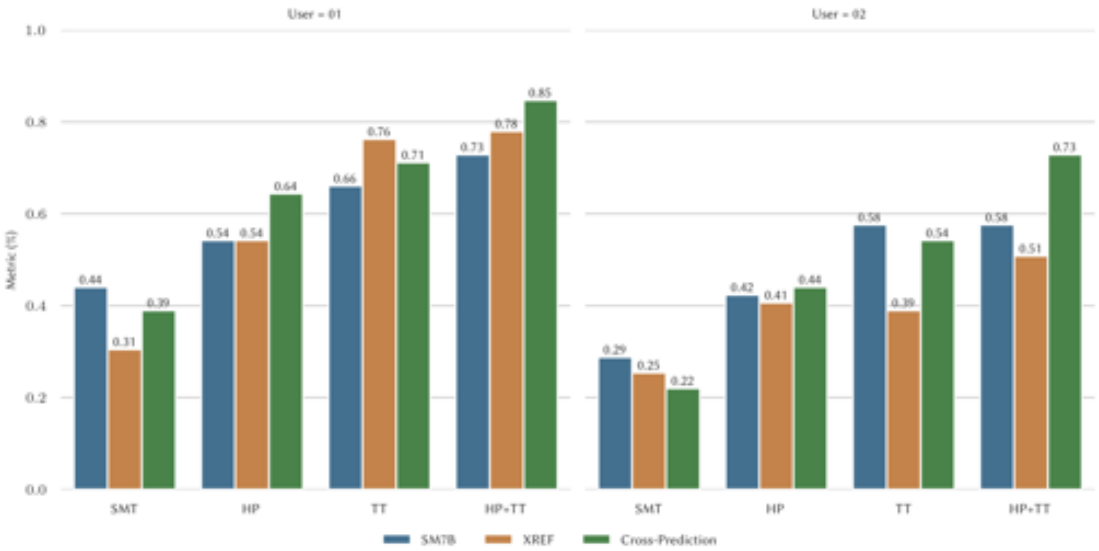


Figure 20: *Single character classification accuracies for user-typed sample text, compared against various combinations of user-created datasets.*

from other users' typing semantics - User 1 who has similar typing semantics to the main training dataset provides better classification using a combination training and personal data, whereas User 2 provides better classification when only combining personal data due to stylistically dissimilar typing mannerisms to the main training dataset.

7.5 Deciphering Contextual Sentences

This section focuses on the feasibility of deciphering a passage of words with contextual meaning from multiple users. Users from Section 7.1 were asked to copy a contextual sentence into the keylogger. This keystroke data is then predicted using Random Forest classifiers on multiple training datasets. Data from both microphones are then cross predicted to produce the classification metrics in Figure 20. The bar plots displayed in Figure 20 show that a combination of users *HP* and *TT* datasets provide the best keystroke recovery rate for both users, presenting cross-prediction accuracies as high as 85% and 73% for User 1 and User 2 respectively. This is the largest recovery rate of keystrokes for user experiments conducted in this paper, which proves the hypothesis that different users emit a specific sonic fingerprint that cannot be accurately interpreted when using generalised datasets such as the initial training dataset used for feature extraction.

To display the output from the cross prediction algorithm, the accuracies for the User 1 and User 2 *HP & TT* datasets detailed in Figure 20 are mapped to their respective contextual sentence below. It should be noted that the majority of misclassified keystrokes are confused with keys situated in similar geometric positions to the ground truth label (e.g., *'r-t'*, 'a-s'), indicating a user produces comparable emanations from neighbouring key classes.

Actual. `when boxing made its olympic debut only one country took all the medals`

User 1. `woen noxing mafe iys olympic debit only one country tooo sll the mefald`

User 2. `wdjn boxibg mage iee olympic debit onpy one counuru uook aol rhe mqdals`

8 Conclusion & Future Work

This paper demonstrates the possibility of a keystroke emanation attack utilising both acoustic emanation and geometric features to classify keystrokes. *MFCC* and *TDoA* features can be combined to achieve inter-dataset keystroke recovery rates for a 40-way classification task of as high as 82% using a Random Forest classifier on a singular datasets. The use of cross-predicting two datasets can be utilised to increase overall detection accuracy up to 89% using supervised inter-dataset analysis, on the condition that the respective datasets possess ample recovery metrics when classifying in solidarity. The use of cross-predictions in this work allows the recovery rate of keystrokes to be higher than that shown in previous research utilising acoustic emanation and geometric features (Zhuang et al., 2009; Halevi & Saxena, 2015).

User experiments were conducted to display real-world attack scenarios. These experiments used 27-way classification models to decipher contextual sentences. When considering intra-dataset analysis on a user not previously trained, it was shown

that a generalised keystroke database provided unsatisfactory recovery rates when users typed using *Touch Typing* mannerisms (52% accuracy for User 1 and 25% accuracy for User 2), even when using cross-prediction methods. However, deciphering contextual arguments using a user-trained dataset allowed recovery rates of up to 85% for User 1 and 73% for User 2, which allowed accurate reconstruction of the sentence. It is hypothesised that a user emits distinct sonic fingerprints from other users even when typing on the same keyboard, but a user also emits distinct keystroke emanations per key class.

The significance of this work is that keystroke emanations are increasingly becoming a security concern and can threaten an unsuspecting user if training data is able to be discretely captured. Mitigation strategies include the use of noise-induced environments to suppress the emanation of keystrokes, or even changes to the design of keyboards themselves so that all keys on a keyboard produce consistent emanations. A promising avenue for producing consistent keystrokes could be haptic technology, which is already adopted by many modern phones and laptop touch pads, but would allow constant emanations without the loss of feedback given to a user.

Future work concerning keystroke emanations would be in the development of a linguistic model to classify phrases with contextual meaning to a degree of greater accuracy. Analysis of keystroke emanations from contextual arguments shows that the majority of misclassified keystrokes reside near the same geometric position as its ground truth label. Linguistic analysis could potentially be used to mitigate this form of inaccuracy. Linguistic analysis has previously been researched by X and Y regarding keystroke acoustic emanations, but they had not considered cross prediction methodologies as shown in this research to improve classification results. Another route of research would be to develop a linguistic model that takes advantage of software-specific semantics - in this scenario, an attacker would utilise specific software key-bindings and key commands (for example, key commands used in command line interface software) to contextually decipher a targets' key presses (Zhuang et al., 2009).

References

Asonov, D., & Agrawal, R. (2004). Keyboard acoustic emanations. *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*, 3–11. https://doi.org/10.1109/SECPRI.2004.1301311

Berger, Y., Wool, A., & Yeredor, A. (2006). Dictionary attacks using keyboard acoustic emanations. *Proceedings of the 13th ACM Conference on Computer and Communications Security*, 245–254. https://doi.org/10.1145/1180405.1180436

Hiu, A., & Fiona, Y. (2006). Thesis ii keyboard acoustic triangulation attack.

Nisonger, T. (2008). The "80/20 rule" and core journals. *Serials Librarian - SERIALS LIBR*, *55*, 62–84. https://doi.org/10.1080/03615260801970774

Zhuang, L., Zhou, F., & Tygar, J. D. (2009). Keyboard acoustic emanations revisited. *ACM Trans. Inf. Syst. Secur.*, *13*(1). https://doi.org/10.1145/1609956.1609959

Backes, M., Dürmuth, M., Gerling, S., Pinkal, M., & Sporleder, C. (2010). Acoustic side-channel attacks on printers. *Proceedings of the 19th USENIX Conference on Security*, 20. https://dl.acm.org/doi/10.5555/1929820.1929847

Scola, C. F., & Ortega, M. D. B. (2010). Direction of arrival estimation : A two microphones approach.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Halevi, T., & Saxena, N. (2012). A closer look at keyboard acoustic emanations: Random passwords, typing styles and decoding techniques. *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, 89–90. https://doi.org/10.1145/2414456.2414509

Genkin, A., Danieland Shamir, & Tromer, E. (2014). Rsa key extraction via low-bandwidth acoustic cryptanalysis. In J. A. Garay & R. Gennaro (Eds.), *Advances in cryptology – crypto 2014* (pp. 444–461). Springer Berlin Heidelberg.

Zhu, T., Ma, Q., Zhang, S., & Liu, Y. (2014). Context-free attacks using keyboard acoustic emanations, 453–464. https://doi.org/10.1145/2660267.2660296

Halevi, T., & Saxena, N. (2015). Keyboard acoustic side channel attacks: Exploring realistic and security-sensitive scenarios. *Int. J. Inf. Secur.*, *14*(5), 443–456. https://doi.org/10.1007/s10207-014-0264-7

Liu, J., Wang, Y., Kar, G., Chen, Y., Yang, J., & Gruteser, M. (2015). Snooping keystrokes with mm-level audio ranging on a single phone. *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 142–154. https://doi.org/10.1145/2789168.2790122

McFee, B., Raffel, C., Liang, D., PW, E. D., McVicar, M., Battenberg, E., & Nieto, O. (2015). Librosa: Audio and music signal analysis in python. *Proceedings of the 14th python in science conference*, 18–25. https://doi.org/10.5281/zenodo.4792298

Shure. (2015). Cardioid dynamic vocal microphone: Sm7b. Retrieved August 21, 2021, from https://pubs.shure.com/guide/SM7B/en-US

Compagno, A., Conti, M., Lain, D., & Tsudik, G. (2017). Don't skype & type! acoustic eavesdropping in voice-over-ip. *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 703–715. https://doi.org/10.1145/3052973.3053005

Anand, S. A., & Saxena, N. (2018). Keyboard emanations in remote voice calls: Password leakage and noise(less) masking defenses. *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, 103–110. https://doi.org/10.1145/3176258.3176341

Sonarworks. (2018). Sonarworks soundid reference measurement microphone. Retrieved August 21, 2021, from https://www.sonarworks.com/soundid-reference/brand-assets

Lee, R., Kang, M.-S., Kim, B.-H., Park, K.-H., Lee, S. Q., & Park, H.-M. (2020). Sound source localization based on gcc-phat with diffuseness mask in noisy and reverberant environments. *IEEE Access*, *8*, 7373–7382. https://doi.org/10.1109/ACCESS.2019.2963768

MATLAB. (2021). *Matlab documentation - cross correlation*. Retrieved August 25, 2021, from https://uk.mathworks.com/help/matlab/ref/xcorr.html

Appendices

A Unix JSON Format

Listing 1: Unix time stamp data storage for keystroke extraction.

```
{
    "start_time_unix": 1628157296079469000,
    "end_time_unix": 1628157302975556000,
    "events": [
        {
            "event_type": "KEY_DOWN",
            "char": "'h'",
            "delta_time": 861685000
        },
        {
            "event_type": "KEY_UP",
            "char": "'h'",
            "delta_time": 940428000
        }
    ]
}
```