Compromising Clicks: A Deep Learning-Based Acoustic Side Channel Attack on Keyboards

Student Name: J.B.F. Harrison
Supervisor Name: Dr. E. Toreini
Submitted as part of the degree of MEng Computer Science to the
Board of Examiners in the Department of Computer Sciences, Durham University

Abstract—With recent developments in deep learning, the ubiquity of microphones and the rise in laptop ownership due to the COVID-19 pandemic, acoustic side channel attacks present a greater threat to keyboards than ever. Unfortunately, deep learning and laptops remain largely under-explored in the literature as a method of classification and as target devices respectively. This paper presents an implementation of a state-of-the-art deep learning model in order to classify laptop keystrokes. When trained on keystrokes recorded by a nearby phone, the classifier achieved an accuracy of 95%, the highest accuracy seen without the use of a language model. When trained on keystrokes recorded using the video-conferencing software Zoom, an accuracy of 93% was achieved, a new best for the medium. Finally, the presented model was trained on Enigma machine keystrokes, achieving an accuracy of 91%, an improvement of almost 8% over the previous best attempt. Across these three data sets, the model furthers the state of the art while the produced results allow for comparison between target devices, attack vectors and the potential improvement of existing work through the addition of deep learning models.

Index Terms—Machine Learning, Neural Nets, Signal Processing, Security and Privacy Protection, Fast Fourier Transforms.

1 Introduction

Side channel attacks (SCAs) involve the collection and interpretation of signals emitted by a device [28]. Such attacks have been successfully implemented utilising a number of emanation types, such as electromagnetic (EM) waves [32], power consumption [16] as well as sound [2]. With such a wide range of available mediums, target devices have been similarly varied, with compromised devices including printers [3], the Enigma machine [30] and even Intel x86 processors [35].

One such target device presenting a looming vulnerability in modern research is the computer keyboard. It was found in [32] that wireless keyboards produce detectable and readable EM emanations, however there exists a far more prevalent emanation that is both ubiquitous and easier to detect: clicks [25]. The ubiquity of keyboard acoustic emanations makes them not only a readily available attack vector, but also prompts victims to underestimate (and therefore not try to hide) their output. For example, when typing a password, people will regularly hide their screen but will do little to obfuscate their keyboard's sound.

It is possible that the cause for the general lack of concern regarding keyboard acoustics is the relatively small body of modern literature. While multiple papers have created models capable of inferring the correct key from test data, these models are often trained and tested on older, thicker, mechanical keyboards with far more pronounced acoustics than modern ones, especially laptops. It is, however, worth considering that while keyboards have gotten less pronounced over time, the technology with which their acoustics can be accessed and processed has improved dramatically.

An example of this can be found in Deep Learning (DL). DL is a subsection of machine learning (ML), in which the model consists of multiple layers of connected neurons. Input data is passed into the model's input neurons, which calculate values in response. The values of these input neurons are then input into the following layer, continuing through layers until the values of output neurons are interpreted as a result. This result is then compared to the truth and consequently the connections between neurons are adjusted to encourage correct behaviour.

Despite being prevalent in the field of computing since the 1960s, DL saw a boom in research in the 2010s benefiting from improvements in graphics processing technology and resulting in huge advances in image recognition [17], the invention of Generative Adversarial Networks [11] and the invention of transformers [31]. This boom in performance continues still, with the recent development of the state-of-the-art CoAt Network for image recognition [8], which combines more traditional convolutional models with transformers. This improvement in DL performance coincides with an increase in access to DL tools. Python packages such as PyTorch [24] provide free and near-universal access to the tools required to run these models on most devices.

In another example, recent research has capitalised on advancements in microphone technology, with VoIP calls [7] and smartwatches [19] being used to collect keystroke recordings. These modern attack vectors on acoustic emanations are not only more capable than those used in prior research, but are additionally more discrete. For comparison, the microphones used in [2] had thick, obvious wires running to the attacker's device, whereas in recent times, a high-quality microphone array can be found in most smartphones, smart-speakers and are ubiquitous to the point of not being suspicious.

With the recent developments in both the performance

of (and access to) both microphones and DL models, the feasibility of an acoustic attack on keyboards begins to look likely, as reiterated in recent research [4]. Another development in recent years contributing to the possibility of acoustic side channel attacks (ASCAs) is the popularity of laptops, which saw a rapid increase during the COVID-19 pandemic.

While recent papers have explored the viability of ASCAs on laptop keyboards [4, 7], the area remains under-explored considering that laptops make a prime attack vector. Laptops are more transportable than desktop computers and therefore more available in public areas where keyboard acoustics may be overheard, such as libraries, coffee shops and study spaces. This lies in contrast to desktop computers with plug-in keyboards (the focus of much of the literature) which would require access to the victim's location to record. Moreover, laptops are non-modular, meaning the same model will have the same keyboard and hence similar keyboard emanations. This uniformity within laptops could mean that, should a popular laptop prove susceptible to ASCA, a large portion of the population could be at risk.

This paper aims to show the viability of DL models in this field as well as the feasibility of an ASCA on modern laptops by answering the research questions:

1) *"Do modern laptop keyboards differ in susceptibility to acoustic side channel attacks compared to older mechanical devices?"*
2) *"Can state-of-the-art deep learning models improve upon existing attacks?"*
3) *"Could an acoustic side channel attack be performed on modern laptops?"*
4) *"If so, how does the accuracy of the attack vary with recording device?"*

In order to answer these questions, the objectives shown in table 1 were defined and in fulfilling these objectives, this paper presents a number of contributions to the field of keyboard acoustic side channel research:

- A new state-of-the-art accuracy when classifying keystrokes recorded via video conferencing app (93%),
- A new state-of-the-art accuracy when classifying keystrokes without using language models (95%),
- A new state-of-the-art accuracy when classifying keystrokes made on an Enigma machine (91%),
- The first use of mel-spectrograms as input features,
- The first use of neural networks in classifying laptop keystrokes,
- The first use of self-attention transformer layers in the field,
- The first comparative study of a classification model on both modern and early-1900s devices.

The remainder of this paper is structured as follows: Section 2 discusses the existing literature on this topic and how this paper presents results relevant to current research, section 3 describes the methodology used to implement the necessary DL models and data sets to fulfill these objectives, section 4 presents the results obtained from following this methodology and discusses their meaning and section 5 concludes the paper reiterating findings and avenues of potential future research.

TABLE 1
The objectives of this paper

Objective No.	Objective	Priority
1.	Process and prepare a labelled data set of keystrokes made on an Enigma machine.	1
2.	Implement the CoAt deep learning model on Enigma keystrokes.	1
3.	Train and evaluate the model against results of existing work in [30].	1
4.	Process and prepare a labelled data set of keystrokes made on a 2021 MacBook Pro.	1
5.	Train a CoAt model and evaluate against the performance of similar approaches in the literature.	1
6.	Create a data set of keystrokes on the same laptop, but recorded via Zoom.	2
7.	Train and evaluate the model on data recorded via Zoom.	2
8.	Perform an offline experiment in which a side channel attack is simulated.	3

2 Related Work

While they remain a relatively under-explored topic of research, ASCAs are not a new concept to the field of cybersecurity. Encryption devices have been subject to emanation-based attacks since the 1950s, with British spies utilising the acoustic emanations of Hagelin encryption devices (of very similar design to Enigma) within the Egyptian embassy [33]. Additionally, the earliest paper on emanation-based SCAs found by this review was written for the United States' National Security Agency (NSA) in 1972 [10]. This governmental origin of ASCAs creates speculation that such an attack may already be possible on modern devices, but remains classified. [2] notes that classified documents produced by the NSA's side channel specification (TEMPEST) are known to discuss acoustic emanations. Additionally, the partially declassified NSA document NACSIM 5000 [21] explicitly listed acoustic emanations as a source of compromise in 1982.

Within the realm of public knowledge, ASCAs have seen varying success when applied to modern keyboards, employing a similarly varied array of methods. Surveying these methods, various observations may be made about the current research landscape.

2.1 The perception of neural networks in ASCAs

In much of the literature, neural networks are not perceived as very successful models when conducting keystroke recognition. In [37], a neural network was tested against a linear classifier and was deemed less accurate. Additionally, in [13] a neural network was found to perform the worst out of all methods tested, and it is noted that neither [37] nor [13] could reproduce the results achieved in [2] through use of a neural network. [1] found that multiple methods performed better than neural networks in testing, while [30]

implemented a neural network that performed third best out of all tested classifiers. A majority of these papers give very little detail regarding the structure or size of the neural networks implemented, making comparison between them difficult, but in none of these cases was a neural network selected as the final model. It is worth noting however that the most recent of the papers to implement a network was written in 2015. Given that Transformers were invented in 2018 by Vaswani et al. [31], this paper is (to our knowledge) the first use of neural networks featuring self-attention layers for an acoustic side channel attack on a keyboard.

2.2 Language models

One approach that saw prominent usage in the 2000's but has become less common in modern papers is the use of hidden Markov models (HMMs). A HMM (in this context) is a model trained on a corpus of text in order to predict the most likely word or character in the positions of a sequence. For example, if a classifier output 'Hwllo', a HMM could be used to infer that 'w' was in fact a falsely classified 'e'. [37] presents a method of ASCA attack on keyboards in which 2 HMMs are utilised: the first generating likely letters from a series of classes and the second correcting the grammar and spelling of the first. Despite being one of the earliest implementations of HMMs in a keyboard ASCA, [37] concluded that performance was greatly improved with their usage, achieving up to 96% character recognition with consistent performance across multiple keyboard models. This state-of-the-art performance was then echoed a year later by a study in which a HMM was used in attacking a dot-matrix printer [3]. Similarly to [37], [3] used a HMM to correct the output of a classifier and saw an increase from 72% to 95% accuracy when implemented. A difference in the two studies however, sheds light on a potential drawback to HMM usage (and the possible reason for lack of recent popularity). While [37] feeds the labels produced by the HMMs back into the classifier, forming an unsupervised training loop, [3] simply outputs the HMM's result. The implication of this is that the method from [3] could never predict random passwords or words unknown to the training corpus, meanwhile the model from [37] could in theory reach a point at which the classifier could recognise individual keys well-enough to ignore the HMMs.

This pattern of performing well for known words but poorly for random ones is not isolated to methods involving HMMs. In fact, any language-based model (also known as context-dependent models) behave similarly. For example, when performing recognition on words from a predecided dictionary, the method in [6] reconstructed 73% of 7-13 character words from recordings of under 5 seconds. Unusually (and perhaps uniquely in the field), performing better on longer sequences of letters. However, similar to [3], the constraint satisfaction algorithm used could only ever produce words from the preset dictionary. The assumption therefore underpinning language-based methods is that the attacker knows not only the language and context of the victim's typing (so as to train HMMs on relevant corpora) but also that a majority of the data being typed will be coherent, grammatically correct sentences.

2.3 Target keyboards

Alongside models, variety exists between studies with respect to target devices. [2], the paper most commonly cited as the first ASCA targeting a keyboard, was written in 2004 and attacked high-profile plastic keyboards synonymous with the time. Despite being such an early paper in the field, success was found in attacking an ATM keypad, a corded telephone as well as 2 keys from a laptop keyboard.

While [37] and [6] perform their experiments on keyboards similar to those from [2], [13] investigates a more modern keyboard with a slightly recessed design. The keycaps remain large and plastic however and differ greatly from modern laptop keyboards. [13]'s authors do however acknowledge that the testing of laptop keyboards may produce different results, due to a lack of 'release peak' in the waveform.

The attack presented in [36] is undertaken on a keyboard similar to that of a modern laptop. The low-profile, predominantly metal keyboard is noted to have an extremely prominent 'hit peak' and inconspicuous 'touch' and 'release' peaks. Such an observation echoes the findings of [13] and this paper, and further hints that such keyboards may be more difficult to classify by sound alone. Despite the impressive accuracy achieved (72.2%), the method used in [36] uses time difference of arrival to calculate the geometric position of keys, and is therefore difficult to compare to the approach taken in this paper.

Of the surveyed literature, [4] and [7] were the only 2 papers to feature ASCAs on full laptop keyboards and are (in the opinion of this paper) the most promising studies with respect to real-world implementation. Both papers utilise two statistical models used in similar ways: the first to infer some information regarding the victim's environment and the second to classify keystrokes into letters. The two papers differ in most other ways however, with [7] gathering keystrokes via Skype and the inbuilt microphone of the laptops, while [4] utilises a mobile phone placed near the victim's computer. Additionally, [7] uses k-NN clustering and a Logistic Regression classifier while [4] utilises support vector machines (SVMs). Despite their differences, both papers are notable for their accuracy, with [4] achieving 91.2% in cross validation and 72.25% when attacking unknown victims and keyboards. Meanwhile [7] achieves a top-5 accuracy of 91.7% given knowledge of the victim's typing style. [4] implements it's attack on 2 laptops, made by Alienware and Lenovo respectively and is notable for being the only study to feature membrane keyboards. [7] presents a much more representative study of keyboards, attacking 6 laptops, two of each: MacBook Pro 13" 2014, Lenovo Thinkpad E540 and Toshiba Tecras M2.

In recent studies, success has been found in attacking touchscreen keyboards, applying the rationale that fingers making impact on a screen will cause varying sounds, detectable by the target device, according to the position on the screen [29, 27, 18]. Such research expands the space of vulnerable devices considerably.

2.4 Keystroke isolation

Patterns within the literature regarding feature extraction and keystroke isolation are discussed in detail throughout

this paper (see subsection 3.1) and so will remain somewhat excluded from this section. However, this paper recognises that is a vital achievement that processes have been defined for the extraction of keystrokes from recordings. Without such a development, the sophistication and performance of many attack strategies simply could not exist. Upon examining multiple pieces of literature however, it becomes clear that across multiple papers (including this one), a similar method of energy-based keystroke isolation is utilised [37, 6, 13, 4]. This method is found to work well for sufficiently loud keystrokes made in an adequately quiet room, however none of the studies surveyed used a method more robust to environmental noise. This lack of robustness limits the capability of ASCAs since if keystrokes cannot be isolated successfully, a majority of methods surveyed would be impossible to implement.

2.5 Attack vectors

With recent years the number of microphones within acoustic range of keyboards has increased and will likely continue to do so. In an attempt to explore these attack vectors, recent research has been utilising alternate methods of keystroke collection. As an example, [36] Implemented an attack utilising a number of off-the-shelf smartphones. These devices (as is the case for a majority of modern phones) feature 2 distinct microphones at opposite ends of the phone. When used together, recordings made by the collective microphones provided sufficient time difference of arrival (TDoA) information to triangulate keystroke position, achieving over 72.2% accuracy. [4] built upon this research by implementing TDoA via a single smartphone in order to establish distance to a target device, eventually achieving 91.52% keystroke accuracy when used within a larger attack pipeline.

Alongside smartphones, video conferencing applications have seen promising results as an attack vector. Keystrokes intercepted from a VoIP call were used in [1], achieving a keystroke accuracy of 74.3% and this success was echoed by [7] which achieved a top-5 accuracy of 91.7% via simply calling a victim over Skype. These successes mark the first ASCAs implemented without the need for physical access to a victim's vicinity and carry the implication that if a victim's microphone could be accessed covertly, a similar attack could be performed.

The same implication can be found with the use of smartwatches as an attack vector. While it remains unlikely an attacker could covertly place their smartwatch in a private location such as an office, compromising a victim's smartwatch could allow unbridled collection of acoustic keystroke information. Additionally, smartwatches can uniquely access wrist motion, a concerning property which is utilised by [19] to achieve 93.75% word recovery.

This exploration of attack vectors other than simple desktop microphones leads to a few key insights, chief among them being the consistent success of such methods. Of the papers surveyed, none saw an accuracy of less than 70% irrespective of recording device, implying that as microphones become more common, so too do the methods for inferring keystrokes. Another insight is that a compromised microphone within a victim's laptop, desktop, smartwatch or smartphone may be a more dangerous prospect than previously thought: completely bypassing the need for discretion in recording keystrokes. However, this paper notes that should a victim's device already be compromised, keystroke emanations may not be required for obtaining passwords when compared to more direct methods such as keyloggers.

It is the opinion of this paper that video or web-conferencing programs would present the most concerning attack vector, if not for evidence presented in both [1] and [7] that multiple mitigation techniques are both implementable and successful in reducing inference accuracy. It is therefore the recommendation of this paper that such mitigation be implemented as standard in these kinds of software.

2.6 Mitigation techniques

With the successes seen in this field and the potential threat these attacks pose [14], it becomes necessary for research to discuss methods of mitigating or defending against them. Despite not being stated as an explicit means of defense, results from [13] imply that simple typing style changes could be sufficient to avoid attack. When touch typing was used, [13] saw keystroke recognition reduce from 64% to 40%, which (while still an impressive feat) may not be a high enough accuracy to account for a complex input featuring the shift key, backspace and other non-alphanumeric keys. Additionally, a change in typing style may be implemented alongside mitigation techniques presented in other papers and requires no software or hardware component.

The second simple defense against such attacks would be the use of randomised passwords featuring multiple cases. With the success of language-based models in [37, 3, 6], passwords containing full words may be at greater risk of attack. Also, while multiple methods succeeded in recognising a press of the shift key, no paper in the surveyed literature succeeded in recognising the 'release peak' of the shift key amidst the sounds of other keys, doubling the search space of potential characters following a press of the shift key.

As stated in section 2.5, papers [1] and [7] present methods and therefore countermeasures based on Skype calling. [1] implements two sound-based countermeasures: playing sounds over a speaker near the broadcasting microphone and mixing sounds into the transmitted audio locally. Of the two, the second is more discrete and less distracting for the user. Two types of sound were tested, white noise and fake keystrokes, with the latter proving to be more effective thanks to the sophistication of white noise removal algorithms. [7] attempted to disrupt keystroke acoustic features by randomly warping the sound slightly whenever keystrokes were detected, a method which reduced accuracy using FFT features to a random guess, but only slightly inhibited MFCC features.

Of the mitigation techniques for voice call attacks, adding randomly generated fake keystrokes to the transmitted audio appears to be of best performance and least annoyance to the user. However, such an approach must only be deployed when keystrokes are detected by the VoIP software as constant false keystrokes may inhibit usability of the software for the receiver. One potential direction of future research is the automatic suppression or removal of

keystroke acoustics from VoIP applications. Such an implementation would not only defend against ASCAs, but would remove irritating keystroke sounds for the users.

[37] recommends a defence which has proven apt with the progression of time in the form of two-factor authentication: utilising a secondary device or biometric check to allow access to data. As more laptops begin to come with biometric scanners built in as standard, the requirement for input of passwords via keyboard is all but eliminated, making ASCAs far less dangerous. However, as stated in [37], a threat remains that data other than passwords may be retrieved via ASCA.

Perhaps equally as interesting as effective countermeasures are those presented in papers that have lost viability over time. For example, [2] states touchscreen keyboards present a silent alternative to keyboards and therefore negate ASCAs, however in recent studies compromised smartphone microphones have repeatedly inferred text typed on touchscreens with concerning accuracy [29, 27, 18]. Similarly, [37] recommends checking a room for microphones before typing private information. Such a technique is nearly entirely negated by the modern ubiquity of microphones. Such a method would require removal of smartphones, smartwatches, laptops, webcams, smart speakers and many more devices from the vicinity. [7] states that muting their microphone or not typing at all when on a Skype call may defend victims from ASCAs. Such an approach lost some feasibility during the COVID-19 pandemic, at which time a large number of companies began to switch to remote working via video-conference software, necessarily including typing. The diminishing of these countermeasures creates concern that as the prevalence of technology required for these attacks increases, further countermeasures will prove insufficient.

2.7 Keystrokes for authentication

Keystroke acoustics have been found to be useful outside of text retrieval. Promising results have been presented in research pertaining to the use of keystrokes for verification of a user. In [20], 28 users entered an identical 10 digit numeric code 200 times each. From this data, a random forest classifier was trained to achieve 99.97% accuracy in classifying who had typed a given code. Perhaps the most interesting part of this study was the use of only three features: the time between successive 'hit peak's, time between the 'hit peak' and 'release peak' and the time between the release of one key and the pressing of the next. Such features are relatively simple to collect and could be applied to a number of target devices.

These results were then emulated and expanded upon in [22] and [5]. [22] implements both an ASCA and a verification model on a keyboard, achieving an 88% accuracy when identifying the user based on just 6 digits after training on 4 passcodes. Meanwhile, [5] maps the sound of users typing to musical properties (melody, harmony, pitch, etc.) before passing the musical data to both statistical and human classifiers. Interestingly, with the typing properties of the users translated into music, both humans and statistical methods could determine different users with an accuracy of over 81%. Statistical models performed better than human classifiers but both methods present strong evidence of typing behaviour being classifiable.

The implications of this research are interesting in isolation, but concerning when combined with progress made in ASCA research. Such studies imply the possibility of an attacker not only retrieving information typed on a keyboard, but also which user of the keyboard typed it. Such knowledge could add significant risk to computers accessible by multiple people such as those in libraries or offices, which otherwise may add confusion to an attacker as to who typed what. Additionally, multiple papers used the time between keystrokes as a key feature, a property that translates to touchscreens.

2.8 Contributions of this paper

Having surveyed the existing literature on the subject of acoustic side channel attacks on keyboards, this paper extends the field of research in a number of ways (all statements are made based on literature surveyed for the writing of this paper):

1) While much of the literature uses MFCC or FFT features, this is the first use of mel-spectrograms as a feature extraction method for an acoustic side channel attack.
2) While laptops were targeted by 2 of the surveyed papers, this paper presents the first use of a neural network to classify a laptop keyboard's acoustics.
3) With respect to neural networks, this paper presents the first use of self-attention transformer layers in an acoustic side channel attack on keyboards.
4) This paper presents the first comparative study of a keystroke classifier on both a modern keyboard and Enigma cryptography device.
5) This paper improves upon the best accuracy achieved in the literature without the use of a language model,
6) This paper improves upon the accuracy achieved by previous literature classifying keystrokes recorded via video-conference,
7) This paper improves upon previous models used to classify keystrokes from an Enigma machine.

3 Methodology

It is common throughout the existing literature to have three stages to investigating ASCA attacks on a keyboard. These three stages consist of: data collection and preparation, model selection and implementation and finally evaluation and experiments. In order to fulfil the objectives of this paper, these three stages were implemented for each of the three attack scenarios (Enigma machine, laptop recorded by phone, laptop recorded via Zoom). To reflect this process, this section is presented with respect to the three stages and how each was undertaken for each scenario.

3.1 Data Collection and Preparation

In order to fulfil objectives 1-3 and begin to answer research questions 1 and 2, a labelled dataset of Enigma keystrokes was required. Fortunately, this paper was written with access to the labelled data set of Enigma keystrokes produced

Fig. 1. Taken from [30], the Enigma machine from various typing angles as well as it's mechanical components

Fig. 2. Desk setup for recording keystrokes

by Toreini et al. in [30]. This data set takes the form of 160 presses of each of the 27 letter keys on the machine, distributed uniformly across 32 participants of varying gender, body strengths and typing abilities. The recordings were pre-isolated into individual keystrokes and took the form of .wav files with a sample rate of 44100Hz, 32 bits per sample and a single (mono) channel. An image from [30] is presented in figure 1 showing subjects typing on an Enigma machine.

In order to fulfil the remaining objectives of the paper, and to allow for comparison between attack scenarios, similar data sets were required consisting of laptop keystrokes recorded via different means. Research questions 1 and 3 state that the keyboard experimented with should be 'modern' and as per the points made in section 1, a popular laptop keyboard would yield the greatest potential attack vector. Consequently, the model selected for use in this experiment was the MacBook Pro 16-inch (2021) with 16GB of memory and the Apple M1 Pro processor.

This particular laptop not only fulfills the criteria defined in the research questions, but additionally presents a greater-than-normal vulnerability should an attack prove possible.

In being the most recent Apple laptop, it features a keyboard identical in switch design to their models from the last 2 years and potentially those in the future. Additionally, the small number of available models at any one time (presently 3, all using the same keyboard) means that a successful attack on a single laptop could prove viable on a large number of devices. In contrast, the Dell store currently lists 33 purchasable models, a majority of which use different build materials or keyboards. Consequently, an acoustic vulnerability in a single Dell keyboard is unlikely to affect a majority of users.

36 of the laptop's keys were used (0-9, a-z) with each being pressed 25 times in a row, varying in pressure and finger, and a single file containing all 25 presses. For the first data set (referred to as 'phone-recorded data'), these 25 keystroke blocks were recorded on an iPhone 13 mini placed 17cm away from the leftmost side of the laptop on a folded piece of micro-fibre cloth (shown in figure 2). The purpose of the cloth was to remove some desk vibration in the recording (as this would vary based on the type of desk used), instead encouraging the model to learn primarily from acoustics. Recordings were made in stereo with a sample rate of 44100Hz and 32 bits per sample.

For the second laptop dataset (referred to as 'Zoom-recorded data'), keystrokes were recorded using the built-in function of the video conferencing application Zoom. The Zoom meeting had a single participant (the victim) who was using the MacBook's built-in microphone array. The noise-suppression parameter of Zoom was set to the minimum possible ('low') but could not be completely turned off. Before typing, the 'Record on this Computer' button was pressed and after 25 keystrokes the 'stop' button was pressed, producing a .m4a sound recording that was converted to .wav format. As noted in [1] and [7], recording in this manor required no access to the victim's environment and in this case, did not require any infiltration of their device or connection.

Once all presses were recorded, a function was implemented with which individual keystrokes could be extracted. Keystroke extraction is executed in a majority of recent literature [37, 6, 13, 4] via a similar method: performing the fast Fourier transform on the recording and summing the coefficients across frequencies to get 'energy'. An energy threshold is then defined and used to signify the presence of a keystroke. The complete isolation process can be seen executed on an excerpt from the phone data in figure 3. The keystrokes isolated for this data were of fixed length 14400 (0.33s).

Isolating the keystrokes proved more difficult with the Zoom data set. Given the noise suppression present in the Zoom recording, the volume of keystrokes varied massively, making the setting of a threshold value difficult. To bypass this, a loop was implemented in which the threshold was adjusted by increasingly small values until the correct number of keystrokes was found, shown in algorithm 1.

While they didn't require isolation, the Enigma recordings were truncated to remove excess samples outside of the keystrokes. This process involved detecting the peaks in the keystroke sound and isolating the recording around these. The concept of three 'peaks' within keystroke acoustics has been explored in a majority of papers on the topic [2, 6, 13, 36, 4] and is understood to represent the finger hitting the key, the key being fully pressed and then released. These peaks were detected as clusters of local maxima, with the sound then clipped 0.24s before the first peak and 0.48s after the first peak, encapsulating the keystroke. A plot of these local maxima can be seen in figure 4.

Having isolated keystrokes from each data set, a processing pipeline was defined of which the core component would be feature extraction. Multiple methods of audio feature extraction exist and the literature commonly varies

Algorithm 1: Zoom keystroke threshold setting

Result: A set of isolated keystrokes, S
Input: A function for isolating keystrokes $Iso(F, P)$, an initial prominence threshold, P, a recording of keystrokes, F, a step value, s and a target number of keystrokes, T;
Initialisation: An empty list of keystrokes, $S = \{\}$;
while $S.length \neq T$ **do**
  $S = Iso(F, P)$
  if $S.length < 25$ **then**
    $P = P - s$
  end
  if $S.length > 25$ **then**
    $P = P + s$
  end
  $s = s * 0.99$
end

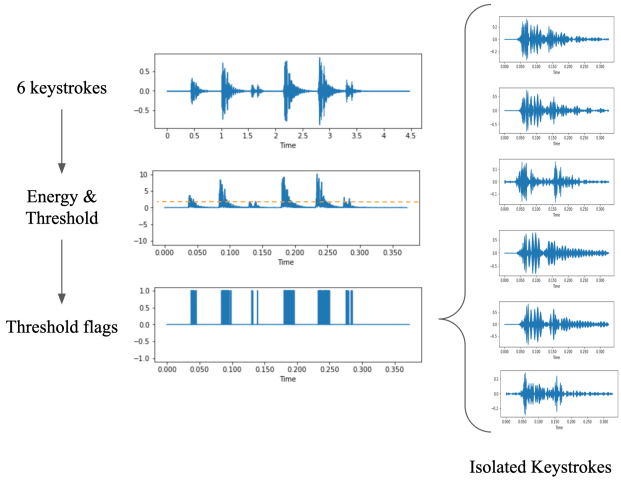

Fig. 3. Keystroke isolation process, signals are converted to energy via FFT, then flagged when crossing the threshold to mark keystrokes

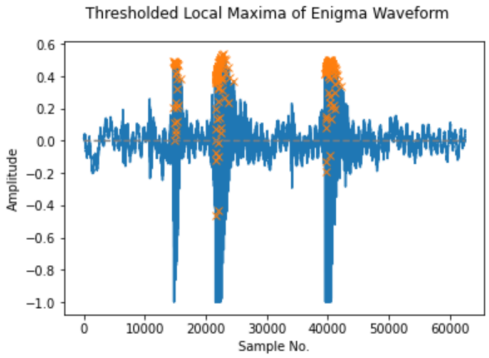

Fig. 4. The local maxima within an Enigma keystroke when thresholded to be above a certain prominence. Clusters of local maxima represent the three peaks of the keystroke

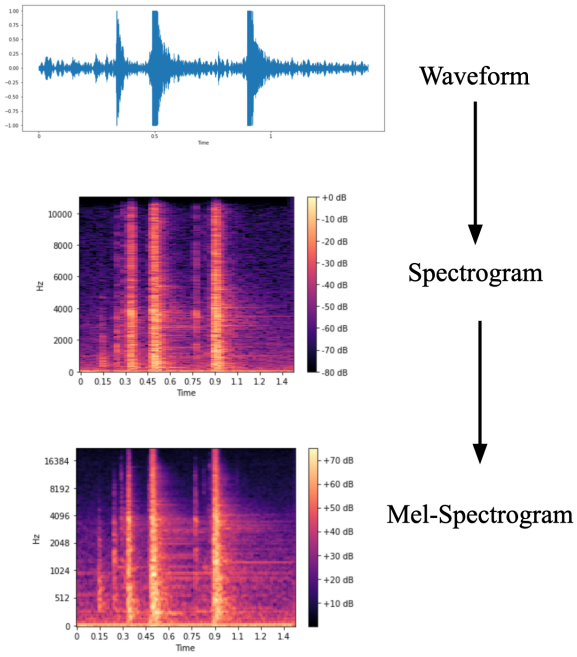

Fig. 5. The visual process of generating a mel-Spectrogram as depicted on an example keystroke form the Enigma machine

as to which is used, however there are common candidates across most ASCA studies:

- The Fast Fourier Transform (FFT)[2]
- Mel-Frequency Cepstral Coefficients (MFCC)[3, 37, 1, 4, 30]
- Cross Correlation (XC)[6, 12]

In this paper, we propose the use of mel-spectrograms as a method of feature extraction for a DL model. A mel-spectrogram is a method of depicting sound waves and is a modified version of a spectrogram. Spectrograms represent sound as a map of coloured pixels, with the Y axis representing frequencies and the X axis representing time. The brightness of a pixel (x, y) in a spectrogram represents the amplitude of a frequency (y) at a given time (x). This concept is then built-upon to form mel-spectrograms, in which the unit of frequency is adjusted to mels: a logarithmic scale more representative of how humans hear sound. A waveform, spectrogram and mel-spectrogram of the same sound can be seen in figure 5.

From figure 5, it is apparent that spectrograms (and specifically mel-spectrograms) represent sound in a visually recognisable manner, a valuable property when considering that DL has been found repeatedly to be one of the best approaches for image classification. For example, in 2021 CoAtNet achieved a state-of-the-art accuracy of 90.88% when classifying 1,000 ImageNet images [8].

It is worth consideration that both FFT and MFCC produce similarly visual depictions of features. In the case of FFT, a spectrogram with linear axis in both frequency and volume is produced. This property makes FFT less suitable for this paper, since a majority of features in keystroke sounds are within the lower frequencies [12, 1, 2] and would therefore be less distinguishable on a linear scale.

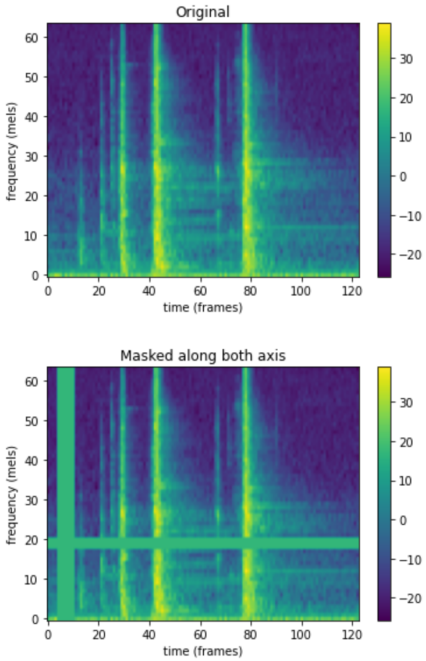

Fig. 6. An Enigma keystroke mel-spectrogram before and after masking

Meanwhile, MFCC involves performing the discrete cosine transform on a mel-spectrogram, producing a compressed representation that prioritises the frequencies used in human speech. Since, for this paper, human speech is not the target, and the removal of frequencies could risk the loss of relevant data, MFCC was decided to be less suitable than mel-spectrograms.

Prior to feature extraction, signals were time-shifted randomly by up to 40% in either direction. This time shifting is an instance of data augmentation, in which the amount of data input to a DL model is artificially increased by slightly adjusting existing inputs [26].

The mel-spectrograms were then generated using 64 mel bands, a window length of 1024 samples and hop length of 500 (255 for the MacBook keystrokes, given their shorter length), resulting in 64x64 images. Using the spectrograms, a second method of data augmentation was implemented called masking. This method involves taking a random 10% of both the time and frequency axis and setting all values within those ranges to the mean of the spectrogram, essentially 'blocking out' a portion of the image as shown in in figure 6. Using time warping and spectrogram masking combined is called SpecAugment and was found to encourage the model to generalise and avoid overfitting the training data [23, 9].

Having converted keystrokes from each data set into a more visual medium, more direct comparisons could be made. Keystrokes recorded on the MacBook are noticeably shorter in length and smaller in amplitude than those of Enigma. Such a difference may contribute to a more robust defense against ASCAs, a useful consideration when answering research question 1. Additionally, MacBook keystrokes (similar to the keystrokes examined in the literature [2, 37, 4]) have only 2 visible peaks: the 'push' and 'release' peaks respectively. Meanwhile the Enigma

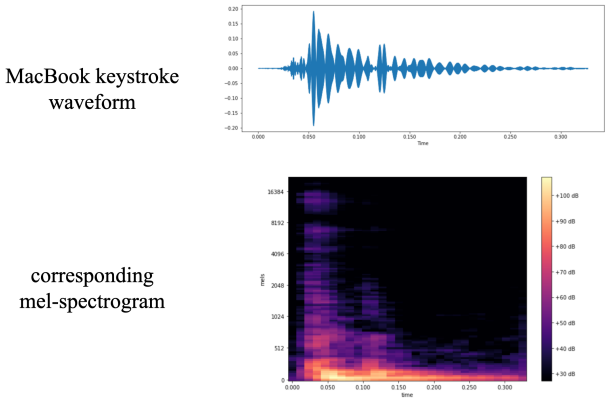

Fig. 7. Phone recording waveform and corresponding mel-spectrogram

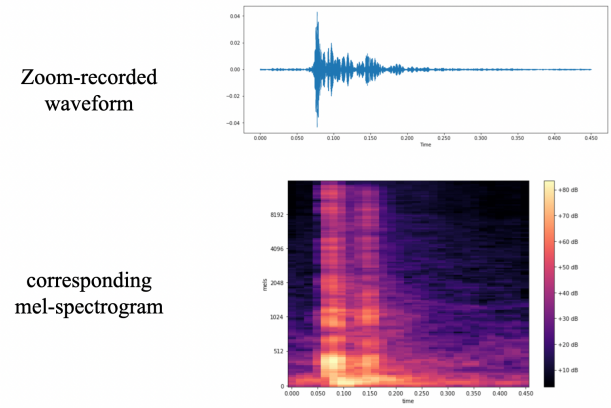

Fig. 8. Zoom recording waveform and corresponding mel-spectrogram

keystrokes have 3 distinct peaks (seen in figure 4).

A phone-recorded keystroke and a corresponding mel-spectrogram can be seen in figure 7, while a Zoom-recorded keystroke is shown in 8.

The 2 peak structure shown in figure 7 is similar to that of 8, implying that such a structure is native to the MacBook keyboard regardless of recording method, a noticeable difference however is the large range of frequencies present in the zoom recording. Similar to the peaks seen in the Enigma data, the Zoom peaks extend much higher than that of the phone-based recordings, indicating significant data in multiple frequencies that were not present when recorded via phone. Given that the Zoom spectrograms show properties present in both the phone and Enigma recordings, it remains as future research to see whether an adjustment of parameters may produce spectrograms more closely resembling one or the other.

The overall data preparation procedure for the Enigma data was based on the structure presented in [9] and is shown in figure 9.

3.2 Model Selection and Implementation

In order to fulfil research questions 2 and 5, a deep learning model was implemented on the processed data. Given

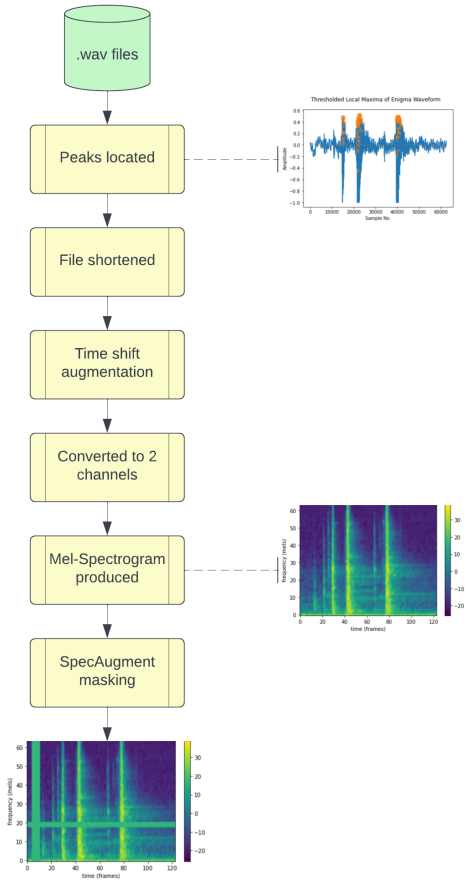

Fig. 9. The data processing pipeline for Enigma data

the visually-distinguishable nature of mel-spectrograms, a model proven to work well for image classification was needed. The CoAtNet model created in [8] was selected due to its excellent performance on the ImageNet classification data set and it's far lower training time compared to similarly performing models.

The mathematical complexities of CoAtNet and the layers within are beyond the scope of this paper, however, from a high-level CoAtNet can be seen to consist of two depthwise convolutional layers followed by two global relative attention layers. The act of combining convolution and self-attention methods allows for rapid processing of patterns in the data while down-sampling the size (convolution) before determining the relevance of these patterns to one another through the calculation of attention scores (self-attention) [8].

In order to implement an instance of CoAtNet, it was decided that PyTorch [24] would be used. PyTorch is an 'open source machine learning framework' that may be installed and used via the programming language Python. Such packages for machine learning are increasingly prevalent in the field and enable standard-issue hardware to run state-of-the-art ML models. Utilising such a package for this investigation highlights the potentially low hardware cost and technical ability required to perform a real-world attack.

The code adapted to suit this particular implementation of CoAtNet was sourced from a public GitHub repository [34]. This repository presents a number of attention-based DL models implemented using PyTorch and based on published papers. Once again, the ease of access to complex DL models (despite requiring understanding in order to adapt them to specific use-cases) validates the potential threat of a DL-based ASCA. All code used for the purposes of this paper was implemented and run in Python 3 within a Jupyter Notebook [15].

The code presented in [34] returns probabilities given in dimensions far greater than the desired shape and number of classes required for this project. To overcome this implementation issue, the output of CoAtNet was reduced to a percentage probability relating to each of the keys. The output of the final self-attention layer was therefore subjected to a 2D average pool followed by a fully-connected linear layer. These additions not only produced the desired output but more closely reflect the desired implementation structure of CoAtNet presented in [8], which can be seen in figure 10.

The initial parameters used for the implemented models were based on those from the original study including the use of the Adam optimiser and cross entropy loss criterion [8].

With the inputs being of identical size, the only parameter requiring adjustment between data sets was the output of the final fully connected layer. Adapting this from the 27 keys of the Enigma machine to the 36 keys of the laptop was sufficient to implement the model on the laptop data.

3.3 Evaluation and Experimentation

This subsection covers the experimental process used to determine appropriate hyperparameters for the classifiers, as well as the method by which the classifiers were evaluated. For the results produced by this evaluation, see section 4.

When implementing a DL model such as CoAtNet, it is important to establish values for various hyperparameters that will define specific behaviours of the model. While some hyperparameters have values found commonly within the literature, a majority of hyperparameter combinations simply have to be tried and validated in order to be compared. One common approach to the hyperparameter optimisation problem is grid search, in which all combinations are tested and the best selected. This method, and those like it, take a large amount of time and, given the already complex model implemented in this paper, repeatedly training and evaluating models would place limitations on the validity of a real-world attack. Ideally, a minimal amount of adjustment would be required to hyperparameters in order to produce a satisfactory model, requiring less time and less competent hardware to execute.

In order to achieve satisfactory model performance with minimal overhead, three hyperparameters were experimented with for each of the three models. The selected hyperparameters were: maximum learning rate (LR), total training epochs and the method of splitting data. The first, LR, represents the rate at which a model's weights adjust to suit the training data, the second defines how many times the model is trained on the entire training data set and the third represents the method used to divide the test, training and validation sets from the overall data pool.

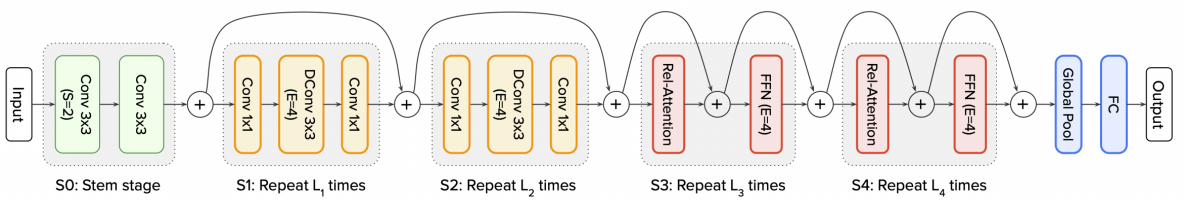

Fig. 10. An overview of the CoAtNet structure, taken from [8]

In order to find appropriate values of these hyperparameters, models were trained then tested against the validation data. The training process of a DL model is well documented and common across most implementations in the literature and such is not covered in depth in this paper. However, a brief overview of this paper's implementation would be as follows:

1) Data was normalised and input to the model in batches,
2) The model produced class probabilities for each item in the batch,
3) These probabilities were used to calculate cross entropy loss and accuracy, with respect to the true values,
4) The loss was used by the optimiser to perform the backpropagation algorithm and adjust the model to better suit the true values,
5) The scheduler was stepped to reduce the learning rate,
6) Every 5th epoch, the model was switched to an evaluation configuration and tested on the validation data.

Utilising such a procedure, models were assigned an LR of 1e-3 and trained for varying numbers of epochs on a stratified split of the data. For each total number of training epochs, the highest validation accuracy seen is logged in table 2.

While the number of epochs required to train a model to a point of convergence varies throughout the literature, the results of this experiment show that this classification requires an uncommonly high number of training epochs when using the default values for LR, momentum (a hyperparameter for the Adam optimiser) and other hyperparameters. The implication of this is that given these default values, the model acquires meaningful interpretation of the data at a relatively slow rate.

A second preliminary experiment was undertaken in which the method of splitting the data was varied. The purpose of this experiment was to provide insights as to the impact of uneven data sets on model performance, as an inconsistent data set is more analogous to a real-world attack. For each dataset, a model was trained for 1100 epochs on an identically-sized training set, split by a either a seeded random or stratified method. The models were then tested for accuracy on similarly-split validation data. The results of the experiment can be seen in table 3.

From table 3 it can be seen that differing the splits of data made little difference to the peak validation set accuracy. Of the three types of data, the Zoom dataset saw the largest difference, increasing validation accuracy by over 13% when using the random split. The other two types of data saw little difference between when trained on different splits, a result that is not surprising. Given the even distribution of classes in all three data sets, random samples of these data sets are expected to be similarly distributed, despite some variance, leading to mostly similar data sets regardless of split. It remains as potential further research as to how models would perform on vastly different distributions of training or testing data.

TABLE 2
The best validation set accuracy seen during training when varying the number of total epochs. LR was set to 1e-3 and all models were trained on a stratified split of data. The number of epochs with highest accuracy for each data set is in bold

Data Set	Total Epochs	Peak Validation Accuracy
Enigma	1300	0.84
	1100	**0.85**
	500	0.77
	300	0.18
	100	0.13
	30	0.05
Phone recordings	1300	0.87
	1100	0.89
	500	**0.92**
	300	0.46
	100	0.29
	30	0.09
Zoom	1300	0.26
	1100	**0.52**
	500	0.14
	300	0.32
	100	0.18
	30	0.03

TABLE 3
Peak validation accuracy achieved when using varying methods of splitting data. Models trained for 1100 epochs with a learning rate of 1e-3

Data	Split	Peak Validation Accuracy
Enigma	Random	0.86
	Stratified	0.85
Phone recordings	Random	0.85
	Stratified	0.89
Zoom	Random	0.59
	Stratified	0.52

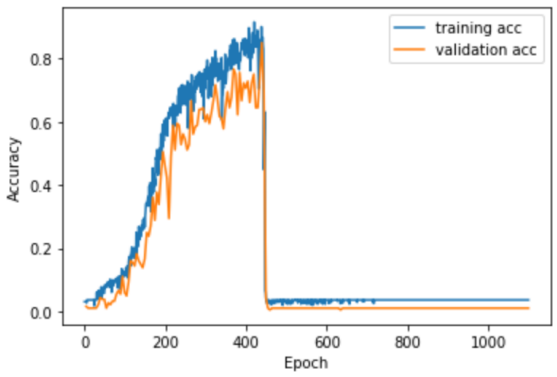

Fig. 11. The training and validation accuracy when training the phone data classifier for 1100 epochs with LR = 1e-3 on randomly split data

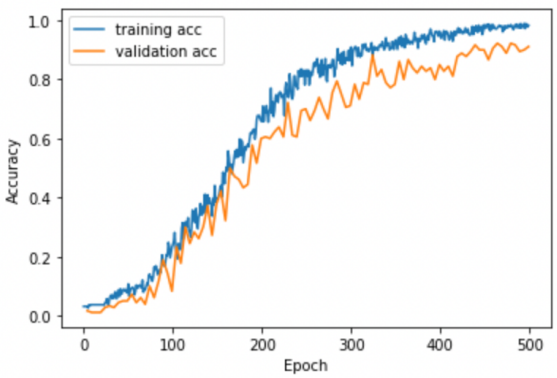

Fig. 13. The training and validation accuracy when training the phone data classifier for 500 epochs with LR = 1e-3 on randomly split data

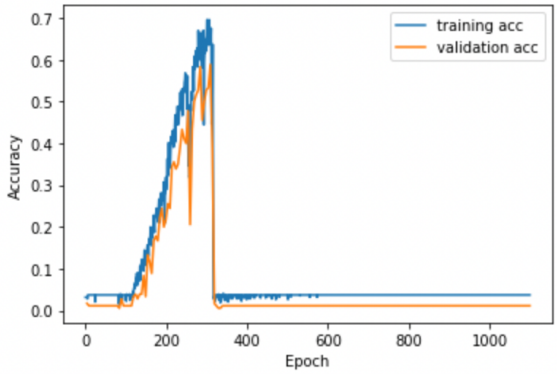

Fig. 12. The training and validation accuracy when training the Zoom data classifier for 1100 epochs with LR = 1e-3 on randomly split data

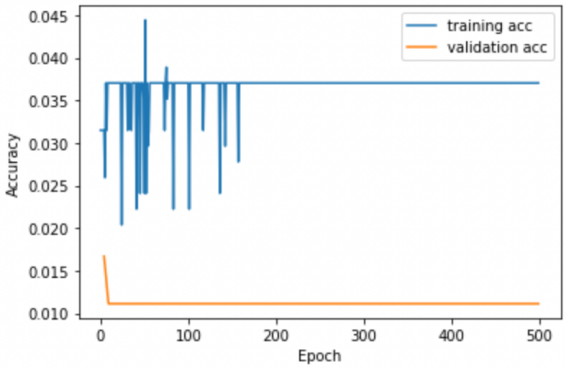

Fig. 14. The training and validation accuracy when training the Zoom data classifier for 500 epochs with LR = 1e-3 on randomly split data

Following the execution of this experiment, the training and validation accuracy was plotted for each of the models to inspect for anomalies, convergence and other behaviours relevant to DL model performance. When examining the accuracy of both MacBook models when trained on randomly split data, an abnormality was found. This abnormality can be seen in figures 11 and 12.

As can be seen from these figures, the models displayed good training progress for a period of 300-400 epochs, before a sudden 'reset' to entirely random prediction. This pattern, while initially achieving a good performance on validation data, clearly indicated a flaw in one or more parameters of the classifiers.

To overcome this issue, both models were trained again on the random split, but for 500 epochs as opposed to 1100, in an attempt to avoid training them past this point of 'collapse'. The results are shown in figures 13 and 14.

While the phone data classifier showed some convergence, as well as a slightly improved peak validation accuracy, the Zoom classifier showed no sign of convergence and performed consistently worse than random on the validation data across all training epochs.

The next step in attempting to address this implementation issue was to adjust the learning rate as opposed to the number of training epochs. In this experiment, each model was trained for the full 1100 epochs, but with an initial learning rate of 5e-4, half the default value. The results of this attempt can be seen in figures 15 and 16.

From these figures, it is apparent that adjusting the learning rate was sufficient to address this issue. By slowing the rate of learning but allowing the models to train for the same number of epochs, not only do they avoid 'resetting' to random, but they also train to a greater accuracy than achieved previously. Following this insight, all 3 parameters were tested in various combinations, with the results reported in 4.

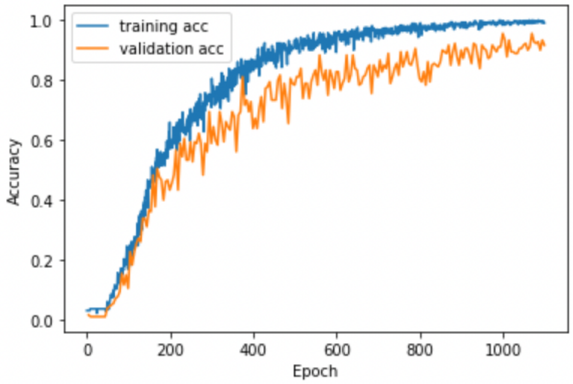

Fig. 15. The training and validation accuracy when training the Zoom data classifier for 1100 epochs with a learning rate of 5e-4

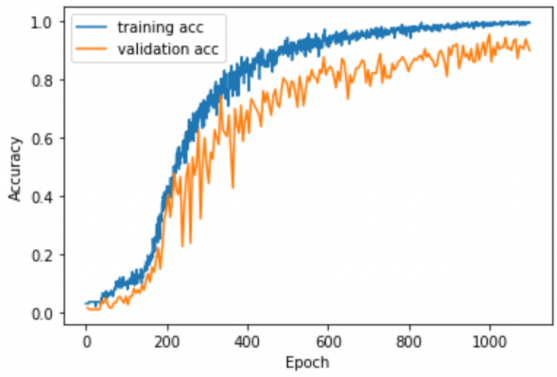

Fig. 16. The training and validation accuracy when training the phone data classifier for 1100 epochs with a learning rate of 5e-4

TABLE 4
Peak validation accuracy achieved when using varying values for all hyperparameters. LR = Learning Rate, PVA = Peak Validation Accuracy

Data	Split	LR	Epochs	PVA
Enigma	Random	1e-3	1100	0.86
	Stratified	1e-3	1100	0.85
Phone	Random	5e-4	1100	0.96
	Random	1e-3	500	0.92
	Stratified	1e-3	500	0.92
	Stratified	1e-3	1100	0.89
Zoom	Random	5e-4	1100	0.96
	Random	1e-3	1100	0.59
	Stratified	1e-3	1100	0.52
	Stratified	5e-4	1100	0.44

From this table, it can be seen once again that for the Enigma and Phone classifiers, the split of data does not appear to be relevant. Meanwhile, the Zoom classifier saw far greater performance when trained on a random split of data with a lower learning rate. Additional values of epoch and learning rate were experimented with, however a validation accuracy of 0.52 could not be improved upon for the stratified Zoom data.

The reason for the Zoom classifier's performance difference across the two splits could be explained by potential anomalous features in the data. Should the recording of a certain key have been effected disproportionately by Zoom's noise reduction feature, inclusion of more instances of that key in the input data could cause confusion in the model's training.

The process of overcoming this issue and consequently experimenting with the three chosen hyperparameters allowed for selection of hyperparameter values for the final models. The values decided for the Enigma and MacBook models can be seen in tables 5, 6 respectively.

Having determined the hyperparameter values to be used, models were instantiated with the desired values and trained on their respective data sets. As in the preliminary experiments, every 5 epochs models were tested on the validation data and the resulting accuracy values were plotted in figures 17, 18 and 19 respectively. Throughout training, cross entropy loss was calculated from the output

TABLE 5
Default hyperparameters used for the model and data processing when training the Enigma keystroke classifier

Parameter	Value
Epochs	1100
Batch Size	250
Loss Type	Cross Entropy
Optimiser	Adam
Max Learning Rate	1e-3
Annealing Schedule	Linear
Waveform Timeshift Percentage	0.4
Spectrogram Mask Percentage	0.1
Number of Masks Per Axis	2
Mel Bands	64
FFT Window Size	1024
Hop Length	500
Data Split	Random
Normalised Data	Yes

TABLE 6
Default hyperparameters used for the model and data processing when training the MacBook keystroke classifiers

Parameter	Value
Epochs	1100
Batch Size	16
Loss Type	Cross Entropy
Optimiser	Adam
Max Learning Rate	5e-4
Annealing Schedule	Linear
Timeshift Percentage	0.4
Max Mask Percentage	0.1
Number of Masks Per Axis	2
Mel Bands	64
FFT Window Size	1024
Hop Length	225
Data Split	Random
Normalised Data	Yes

class probabilities, this loss can be seen to reduce across training epochs in the same figures.

Once trained, all three models were evaluated against their respective unseen test set. This test data was prepared using the pipeline defined in figure 9 except for the time shift and spectrogram masking augmentations: attempting to simulate real-world data. Each model was then input the test data and the resulting predictions were used to generate a classification report and confusion matrix, presented in section 4.

The methodology presented throughout this section is validated by both concurrent and face validity. The evaluation approaches described in this subsection have considerable face validity given the metrics selected (f1-score, precision, recall) being objective measurements of a model's performance on a given test set.

Concurrent validity refers to validity inherited through relation to existing validated tests. In such a manor, the methodology used throughout this paper inherits a consid-

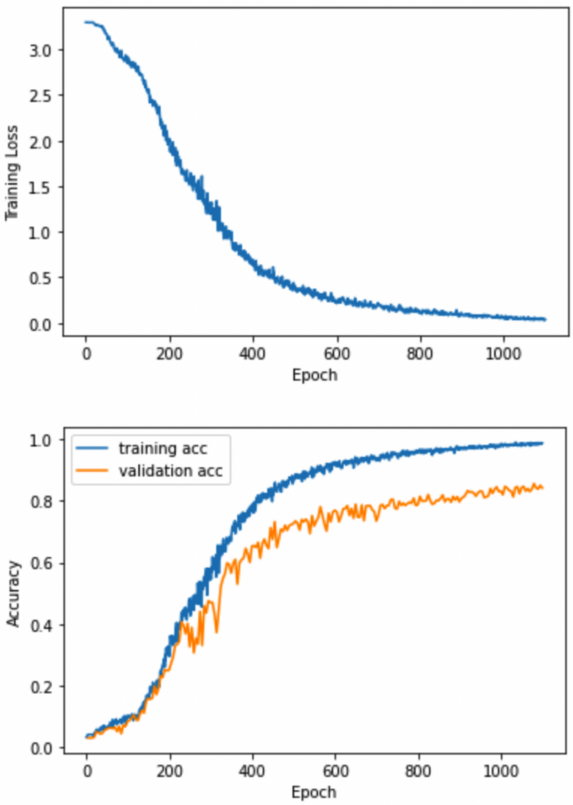

Fig. 17. The testing accuracy, validation accuracy and training loss plotted relative to epochs for the Enigma model

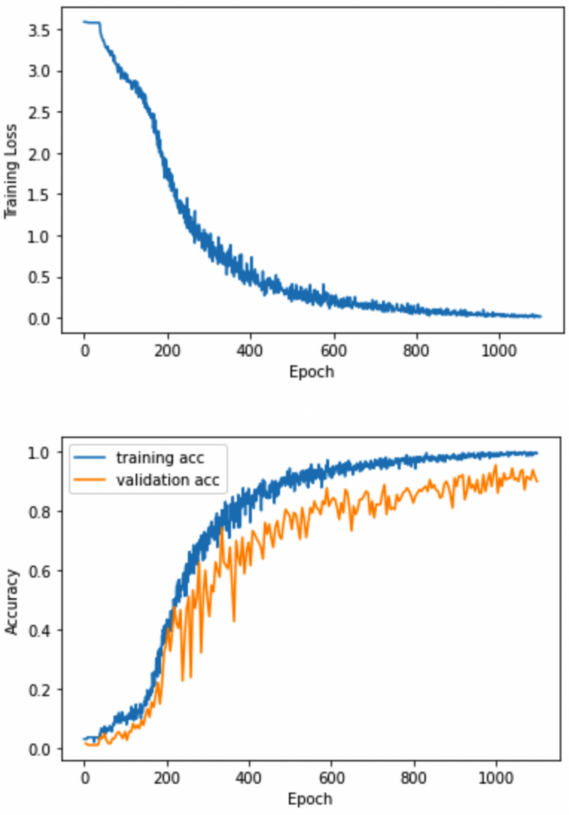

Fig. 18. The testing accuracy, validation accuracy and training loss plotted relative to epochs for the phone recordings model

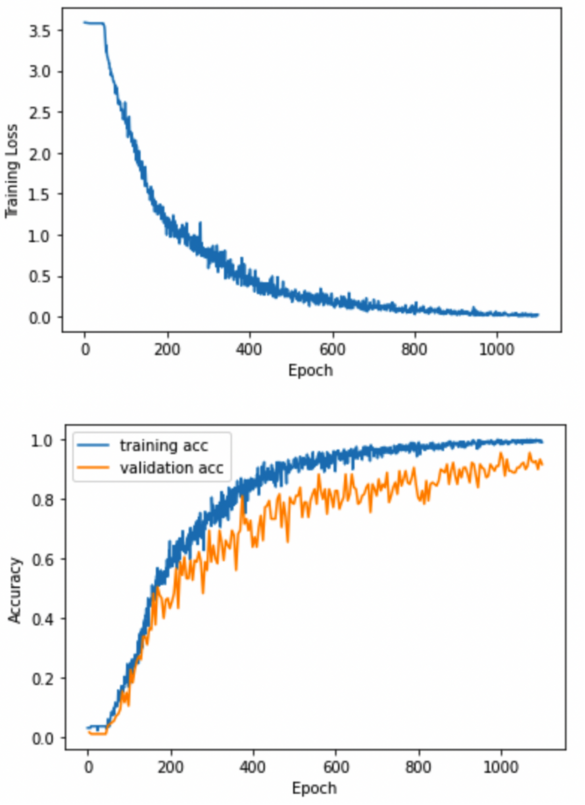

Fig. 19. The testing accuracy, validation accuracy and training loss plotted relative to epochs for the Zoom model

erable amount of validity from the existing literature. While specifically mel-spectrograms were used as input features for this paper, the process of creating mel-spectrograms uses, and is used in, the creation of FFT and MFCC features respectively. Both of these methods are employed successfully throughout the literature [2, 3, 37, 1, 4, 30, 6, 12], lending validity to the use of mel-spectrograms as features. Additionally, the model architecture CoAtNet has seen successful use in image classification [8] and in it's structure uses self-attention layers, an increasingly active component in DL research. As a metric, accuracy is prevalent across the literature [3, 2, 37, 6, 13, 36, 1, 4], as is top-x accuracy (top-5 in this case) [2, 6, 1, 4].

4 Results and Discussion

This section presents the results of the evaluation process defined in section 3.3, and discusses their relevance to the objectives and research questions defined in the introduction. As per this evaluation process, confusion matrices and classification reports were created for each of the classifiers based on test set performance, these are presented in figures 20-22 and tables 7-9 respectively.

A confusion matrix depicts the number of times a model classified an instance of each class on the X-axis as being in a class on the Y-axis. For example, a value of 3 in cell (x, y) where $x = 1$ and $y = 2$ would mean the classifier output 2 for an instance of class 1, 3 times. Therefore, a perfect classifier would result in a confusion matrix with 0 in every cell where $x \neq y$.

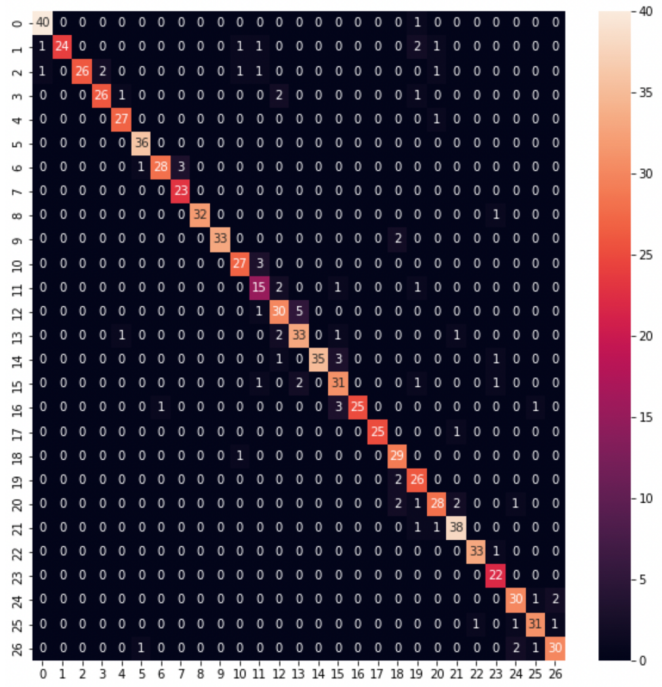

Fig. 20. Confusion matrix for the Enigma keystroke classifier when evaluated on unseen test data

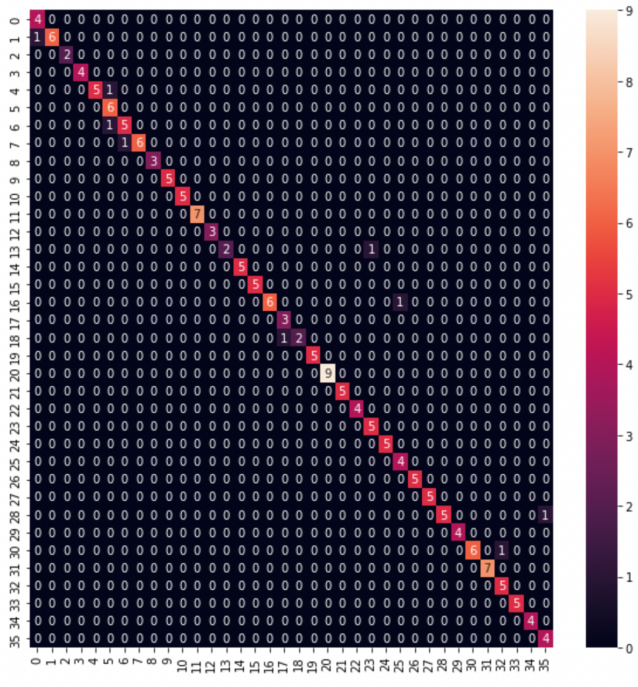

Fig. 21. Confusion matrix for the phone-recorded MacBook keystroke classifier when evaluated on unseen test data

TABLE 7
Classification report for the Enigma keystroke classifier

	Precision	Recall	F1-Score	Support
Accuracy	_	_	0.91	864
Macro Avg	0.91	0.91	0.90	864
Weighted Avg	0.91	0.91	0.91	864

A classification report details the precision, recall, f1-score and support for each class in the data set. Given the large number of classes in all three datasets, we present just the averaged values of these metrics across all classes, as well as the support of the entire test set.

By comparing the performance metrics of the phone recording and Enigma classifiers, an answer can begin to be drawn for research question 1: *"Do modern laptop keyboards differ in susceptibility to acoustic side channel attacks compared to older mechanical devices?"*. Of the two classifiers, the phone recording classifier resulted in a higher precision, recall and f1-score by 0.05, 0.04 and 0.04 respectively. The MacBook proving to be more easily classifiable is unexpected given the low-profile nature of laptop keys, however this could potentially be explained by differences in recording environment or microphone quality. It is worth noting however, that the phone recording classifier was trained on 5 times less data than the Enigma classifier, and so it achieving a greater accuracy implies that the MacBook is more susceptible to some degree.

TABLE 8
Classification report for the phone-recorded MacBook keystroke classifier

	Precision	Recall	F1-Score	Support
Accuracy	_	_	0.95	180
Macro Avg	0.96	0.95	0.95	180
Weighted Avg	0.96	0.95	0.95	180

When comparing the confusion matrices of the two classifiers, two main patterns can be found. In figure 20, there appears the diagonal line of correct classifications typical of a well-performing model as well as clusters of false-classifications within the data. These clusters can be found throughout, such as the 2x2 grid of 1s at (10,1) or the bottom right 3x3 grid. The implication of clustering within a confusion matrix is that there are collections of keys that create features similar to one another. The tendency of these clusters to lie near or on the diagonal line (combined with the way the keys in this experiment were numbered) implies that the position of a key on an Enigma keyboard plays a large role in recognition. For example, the 3x3 grid centred at (12,12) contains 10 false-classifications, all of which were within 1 class of the true answer. This property is also observed in [30], showing consistency across studies and different classifiers.

This notion of position playing a large part in keystroke recognition is reinforced by the tendency for false-classifications to be only a single key 'away' in the phone recording classifier's confusion matrix. In figure 21, $5/9$ false-classifications are a single key away from the true value, and $6/9$ are within 2 keys of the true value. The consistency of this pattern across both classifiers goes some way as to answering research question 1: while the MacBook's keystrokes were found to be slightly easier to classify, both keyboards rely on a similar positional layout and are therefore similarly classifiable when recorded with external microphones. With regards to overall susceptibility to an

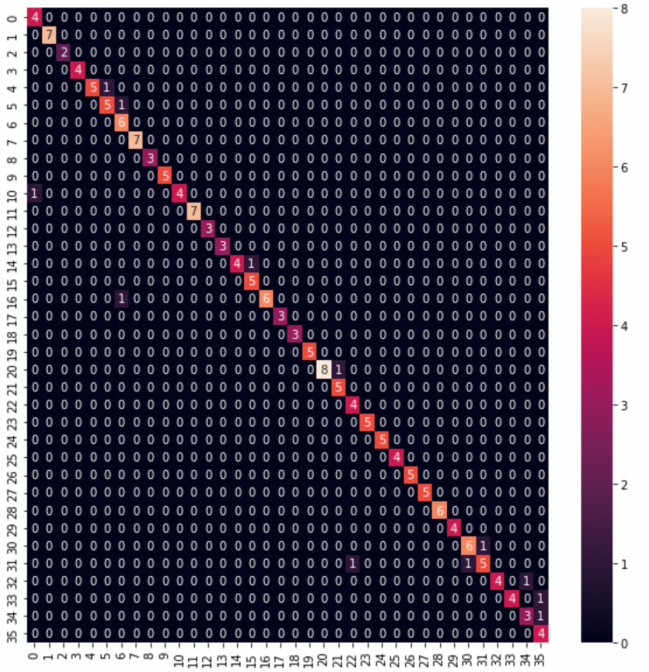

Fig. 22. Confusion matrix for the Zoom-recorded MacBook keystroke classifier

TABLE 9
Classification report for the Zoom-recorded MacBook keystroke classifier

	Precision	Recall	F1-Score	Support
Accuracy	_	_	0.93	180
Macro Avg	0.94	0.94	0.94	180
Weighted Avg	0.94	0.93	0.93	180

attack, the quieter MacBook keystrokes may prove harder to record or isolate, however, once this obstacle is overcome, the keystrokes appear to be similarly if not more susceptible.

The performance of the Enigma classifier is additionally relevant to research question 2. Since a keystroke classifier has been implemented on this exact data in [30], the methodologies are easily comparable via the results they achieved. When classifying the test set, the Discriminant Analysis classifier achieved an accuracy of 84.31% and the implemented artificial neural network achieved 67.14%. The approach used in this paper improved on both the previous study's neural network and their best performing model, classifying the test set with an accuracy of 91% accuracy.

While the data used for both MacBook classifiers was created for this paper, and consequently no other studies have been undertaken on it, the models implemented in this paper were found to have a higher accuracy than all other neural networks surveyed in the literature. Additionally, [7] achieved a top-5 accuracy of 91.7% when classifying MacBook keystrokes via the video conferencing application Skype. When trained on keystrokes from a similar laptop, recorded via a similar medium, the Zoom keystroke classifier presented by this paper achieved an accuracy of 93%.

Insights can be gleaned with respect to research question 4 through the comparison of results for both the Zoom and phone recording classifiers. Of the two recording devices, the precision recall and f1-score were higher by 0.02 for the model trained on the phone data. Such a small difference implies that the utilisation of alternative methods of recording may not necessarily diminish classification accuracy by a noticeable amount. It remains as a potential direction of future research as to whether other discrete methods of recording maintain a similar effectiveness.

While research question 3 requires further experimentation than that performed in this paper to fully answer, the results produced go some way as to indicating the viability of such an attack. Both the phone and Zoom recording classifiers achieved state-of-the-art accuracy given minimal training data in a random distribution of classes. In addition, this paper has shown the effectiveness of both mel-spectrograms as features and self-attention transformers as models when performing keystroke classification. The implementation of such models and feature extraction methods could be used alongside or to replace existing methods in order to further the field.

An observation from the results supporting the possibility of a real-world ASCA is the tendency of each classifier to cluster false-classifications around the correct key. This trait was recognised in [30] and implies that a false-classification may still hold information regarding the location of the true key on the board, a property that could be exploited in future research. However, ultimately, it remains unanswered as to whether an ASCA could be implemented on a modern keyboard; it will likely remain unanswered until one is.

5 Conclusion

This paper has surveyed, identified and investigated gaps in the current literature by attempting to answer the research questions:

1) *"Do modern laptop keyboards differ in susceptibility to acoustic side channel attacks compared to older mechanical devices?"*
2) *"Can state-of-the-art deep learning models improve upon existing attacks?"*
3) *"Could an acoustic side channel attack be performed on modern laptops?"*
4) *"If so, how does the accuracy of the attack vary with recording device?"*

Through surveying and incorporating techniques from relevant literature, a data processing pipeline and deep learning classifier have been implemented with the objective of classifying keystrokes based on their acoustic emanations. The processing pipeline was developed based on the structure presented in [9], utilising mel-spectrograms as features and the SpecAugment method of data augmentation originally presented in [23]. The deep learning classifier was implemented as an instance of CoAtNet, a state-of-the-art image classification model that saw 90.88% accuracy when created by the Brain team at Google research [8].

This paper's overall strategy of data processing and deep learning classifier was used to gain various insights relating to the research questions. To gain these insights, 3 data sets were used for classification: a collection of pre-isolated keystrokes from an Enigma machine produced in

[30] and 2 in-house data sets of keystrokes made on a 2021 MacBook Pro 16': one recorded via an iPhone placed nearby and the other recorded using built-in functionality of the video conferencing software Zoom.

In order to answer the first research question, the methodology presented in this paper was implemented on both the Enigma and phone-recorded MacBook data sets. On the Enigma data, the produced classifier achieved an accuracy of 91%, meanwhile the model trained on the phone-recorded keystrokes achieved an accuracy of 95%. This difference implies that some aspect or combination of aspects of the MacBook keyboard makes it more susceptible to an eavesdropping attack, however given that the two data sets were recorded in different experimental settings, this result is not sufficient to conclusively answer the question. It remains as a topic of future research as to whether MacBook keystrokes continue to be more-easily classified when recorded under identical conditions to an Enigma machine.

The second research question may be answered through comparison of this paper's results with those of previous investigations in the literature. Of the surveyed literature, the highest accuracy achieved for keystroke classification was done so in [37], retrieving 96% of keystrokes. Similarly, the highest accuracy achieved without the use of a language model was 91.2% in [4] and the highest top-5 accuracy achieved when using keystrokes recorded via video-conferencing applications was 91.7% achieved by [7]. On the Enigma machine, the highest accuracy found within the literature was 84.31% presented in [30].

The method presented in this paper achieved a top-1 classification accuracy of 95% on phone-recorded laptop keystrokes, representing a new state-of-the-art result for classifiers not utilising language models and the second best accuracy seen across all surveyed literature. When implemented on the Zoom-recorded data, the method resulted in 93% accuracy, a new state-of-the-art result for classifiers using such applications as attack vectors. Additionally, the presented approach achieved an accuracy of 91% on the Enigma data, an improvement of 7% over the previous best result. These comparisons act as strong evidence that, with regards to this paper's second research question, state-of-the-art deep learning models can improve upon existing attacks.

This paper's third research question is only truly possible to answer conclusively through the real-world execution of an acoustic side channel attack on keyboards. As an end-to-end attack is beyond the scope of this paper, this question remains without a final answer. In spite of this, observations made and results obtained through the writing of this paper do serve as evidence for the viability for such an attack. The introduction to this paper describes the threat landscape for these kinds of attacks and how the variety, accessibility and effectiveness of attack methods is rapidly increasing. This increasing level of threat was evidenced by surveying the related literature, observing recent successful attempts at classifying keystrokes made on multiple devices, based on sounds recorded via smartwatches, smartphones and more. Further to this, a novel approach was undertaken in order to create a keystroke classifier and was used to prove that advances in areas such as deep learning and video conferencing software can lead to more successful classification.

The final research question of this paper is answered in part by a comparison between the performance of classifiers produces for both the Zoom and phone recorded data sets. Examining the data: using an alternative method of recording the keystrokes led to a very minor reduction in accuracy, with the Zoom classifier achieving an f1-score of 0.02 less. This comparison carries an implication that, should an ASCA prove possible on keyboards recorded via nearby microphones, the same attack may prove possible via a number of devices. This implication carries a concerning risk considering the rapidly increasing number of microphones in public places and homes.

In attempting to answer this paper's research questions, the field of acoustically attacking keyboards has been expanded in a number of ways:

- State-of-the-art accuracy was achieved when classifying keystrokes recorded via a video conferencing application (93%),
- State-of-the-art accuracy was achieved when classifying keystrokes without the use of language models (95%),
- State-of-the-art accuracy was achieved when classifying keystrokes made on an Enigma machine (91%),
- Mel-spectrograms were used as input features for the first time,
- A neural network was used to attack a laptop keyboard for the first time,
- Self-attention transformer layers were used for this purpose for the first time,
- A comparison was made between a model's performance on early-1900s and modern laptop keyboards for the first time.

Possible directions of future research include: more robust methods of isolating keystrokes from a single recording, as all ASCA methods rely on accurately isolated keystrokes in order to classify; the use of smart speakers to record keystrokes for classification, as these devices remain always-on and are present in many homes; the implementation of a language model in addition to the method presented in this paper; which could improve keystroke recognition when identifying defined words as well as an end-to-end real-world implementation of an acoustic side channel attack on a keyboard.

REFERENCES

[1] S Abhishek Anand and Nitesh Saxena. "Keyboard emanations in remote voice calls: Password leakage and noise (less) masking defenses". In: *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*. 2018, pp. 103–110.

[2] Dmitri Asonov and Rakesh Agrawal. "Keyboard acoustic emanations". In: *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*. IEEE. 2004, pp. 3–11.

[3] Michael Backes et al. "Acoustic {Side-Channel} Attacks on Printers". In: *19th USENIX Security Symposium (USENIX Security 10)*. 2010.

[4] Jia-Xuan Bai, Bin Liu, and Luchuan Song. "I Know Your Keyboard Input: A Robust Keystroke Eavesdropper Based-on Acoustic Signals". In: *Proceedings of the 29th ACM International Conference on Multimedia*. 2021, pp. 1239–1247.

[5] Amith K Belman et al. "Authentication by mapping keystrokes to music: the melody of typing". In: *2020 International Conference on Artificial Intelligence and Signal Processing (AISP)*. IEEE. 2020, pp. 1–6.

[6] Yigael Berger, Avishai Wool, and Arie Yeredor. "Dictionary attacks using keyboard acoustic emanations". In: *Proceedings of the 13th ACM conference on Computer and communications security*. 2006, pp. 245–254.

[7] Alberto Compagno et al. "Don't skype & type! acoustic eavesdropping in voice-over-ip". In: *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. 2017, pp. 703–715.

[8] Zihang Dai et al. "Coatnet: Marrying convolution and attention for all data sizes". In: *Advances in Neural Information Processing Systems* 34 (2021).

[9] Ketan Doshi. *Audio deep learning made simple: Sound classification, step-by-step*. May 2021. URL: https://towardsdatascience.com/audio-deep-learning-made-simple-sound-classification-step-by-step-cebc936bbe5.

[10] Jeffrey Friedman. "Tempest: A signal problem". In: *NSA Cryptologic Spectrum* 35 (1972), p. 76.

[11] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems* 27 (2014).

[12] Tzipora Halevi and Nitesh Saxena. "A closer look at keyboard acoustic emanations: random passwords, typing styles and decoding techniques". In: *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*. 2012, pp. 89–90.

[13] Tzipora Halevi and Nitesh Saxena. "Keyboard acoustic side channel attacks: exploring realistic and security-sensitive scenarios". In: *International Journal of Information Security* 14.5 (2015), pp. 443–456.

[14] Ajoy Kumar Khan and Hridoy Jyoti Mahanta. "Side channel attacks and their mitigation techniques". In: *2014 First International Conference on Automation, Control, Energy and Systems (ACES)*. IEEE. 2014, pp. 1–4.

[15] Thomas Kluyver et al. "Jupyter Notebooks – a publishing format for reproducible computational workflows". In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. Ed. by F. Loizides and B. Schmidt. IOS Press. 2016, pp. 87–90.

[16] Paul Kocher, Joshua Jaffe, and Benjamin Jun. "Differential power analysis". In: *Annual international cryptology conference*. Springer. 1999, pp. 388–397.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012).

[18] Li Lu et al. "Keylistener: Inferring keystrokes on qwerty keyboard of touch screen through acoustic signals". In: *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE. 2019, pp. 775–783.

[19] Anindya Maiti et al. "Smartwatch-based keystroke inference attacks and context-aware protection mechanisms". In: *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*. 2016, pp. 795–806.

[20] Roy A Maxion and Kevin S Killourhy. "Keystroke biometrics with number-pad input". In: *2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*. IEEE. 2010, pp. 201–210.

[21] NSA NACSIM. "5000: Tempest Fundamentals". In: *National Security Agency* (1982).

[22] Sourav Panda et al. "Behavioral acoustic emanations: Attack and verification of pin entry using keypress sounds". In: *Sensors* 20.11 (2020), p. 3015.

[23] Daniel S. Park et al. "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition". In: *Interspeech 2019*. ISCA, Sept. 2019. DOI: 10.21437/interspeech.2019-2680. URL: https://doi.org/10.21437%2Finterspeech.2019-2680.

[24] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[25] Joseph Roth et al. "Investigating the discriminative power of keystroke sound". In: *IEEE Transactions on Information Forensics and Security* 10.2 (2014), pp. 333–345.

[26] Connor Shorten and Taghi M Khoshgoftaar. "A survey on image data augmentation for deep learning". In: *Journal of big data* 6.1 (2019), pp. 1–48.

[27] Ilia Shumailov et al. "Hearing your touch: A new acoustic side channel on smartphones". In: *arXiv preprint arXiv:1903.11137* (2019).

[28] François-Xavier Standaert. "Introduction to side-channel attacks". In: *Secure integrated circuits and systems*. Springer, 2010, pp. 27–42.

[29] Kai Ren Teo et al. "Retrieving Input from Touch Interfaces via Acoustic Emanations". In: *2021 IEEE Conference on Dependable and Secure Computing (DSC)*. IEEE. 2021, pp. 1–8.

[30] Ehsan Toreini, Brian Randell, and Feng Hao. "An acoustic side channel attack on enigma". In: *School of Computing Science Technical Report Series* (2015).

[31] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[32] Martin Vuagnoux and Sylvain Pasini. "Compromising electromagnetic emanations of wired and wireless keyboards." In: *USENIX security symposium*. Vol. 8. 2009, pp. 1–16.

[33] Peter Wright. "Spycatcher: The Candid Autobiography of a Senior Intelligence Officer". In: *New York: Viking* (1987).

[34] Xiaoma Xmu. *External-attention-pytorch/coatnet.py at master · Xmu-Xiaoma666/external-attention-pytorch*. Oct. 2021. URL: https://github.com/xmu-xiaoma666/External-Attention-pytorch/blob/master/model/attention/CoAtNet.py.

[35] Yuval Yarom and Katrina Falkner. "{FLUSH+RELOAD}: A High Resolution, Low Noise, L3 Cache

{Side-Channel} Attack". In: *23rd USENIX security symposium (USENIX security 14)*. 2014, pp. 719–732.

[36] Tong Zhu et al. "Context-free attacks using keyboard acoustic emanations". In: *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*. 2014, pp. 453–464.

[37] Li Zhuang, Feng Zhou, and J Doug Tygar. "Keyboard acoustic emanations revisited". In: *ACM Transactions on Information and System Security (TISSEC)* 13.1 (2009), pp. 1–26.