Ambient Light Sensors:
Fingerprinting Environments

Daniel Knight

Student Number: 101022997

Submitted as part of the requirements for the award of the
MSc in Information Security
at Royal Holloway, University of London.

Information Security Group

Royal Holloway, University of London

August - 2024

Dissertation Structure

The structure of this document will be as follows:

Contents

List of Figures

List of Tables

Summary

In a world of ever-increasing computational power and connectivity, smartphones have become completely integral and relied upon in our daily lives. Sensor hardware technology has been at the forefront of facilitating the rising popularity through increased functionality. However, as the use and sensors such as gyroscopes, accelerometers and ambient light sensors become more prevalent so too do the security and privacy risks associated with them. This dissertation explores the existing security vulnerabilities and mitigation strategies for smartphone sensors, then this work will focus on the concerns surrounding ambient light sensors and their possibility for location inference through location leakage.

In order to explore these privacy and security concerns of ambient light sensors, research was carried out. Firstly, a comprehensive review of relevant literature surrounding smartphone sensors such as the ambient light sensors and the android stack was carried out. This was done to establish a foundational understanding of the overall risks and possible mitigation strategies that the Android smartphone eco-system faces. Then a Java-based Android application was created to capture the real-world environmental lux data of multiple locations from an android's ambient light sensor, including indoor and outdoor environments. Then the collected data was analysed, through various techniques such a key feature extraction, visual analysis, and machine learning, with the guise of inferring environmental information to find the digital fingerprint,

The analysis performed by this dissertation revealed that while it is possible to use ambient light data to infer a user's location, there are limitations to the results gathered by this project. In that, the sample size taken was too short to allow for environmental distinctiveness to be observed. However, the disparity between outdoor and indoor environments was clearly observed and using classification machine learning, new data was correctly assigned to outside environments. This dissertation also found that the ambient light sensor could easily be exploited by malicious actors on account of android permissionless sensors.

Furthermore, the research performed highlighted issues in existing mitigation strategies and highlighted the need for improved security measures surrounding individual mobile sensors, such as ambient light sensors. The findings of the dissertation show that while there are possible significant PII violations, most notably in determining if a user is in an indoor or outdoor environment, more comprehensive data collection methods and analysis techniques are needed to fully understand the risks and be able to develop mitigations for these risks. Ultimately, the security protocols surrounding ambient light sensors should be reevaluated in favour or stricter permissions to protect user's security and privacy.

Chapter 1: Introduction

1.1 Research Question

This project's research question is" How can smartphone ambient light sensors [1] (ALS) be exploited to infer environmental information and what measures can be taken to mitigate the potential privacy and security risks"

1.1.1 Research Objectives/Aims

The research objectives and aims of this document are as follows:

1. Identify the existing research present, related to the security and privacy concerns of mobile ambient light sensors and its implications to personal security, PII and information inference, such as an environment's digital fingerprint.
2. Design and develop a Java Android application that can be used to gain lux data from a smartphone's ambient light sensor
3. Present and format captured ambient light sensor data for analysis from different common locations, such as household rooms and outside environments
4. Investigate the risk around how a malicious actor would be able to exploit the vulnerabilities of ambient light sensors to gain access to unauthorised personal data
5. To assess and evaluate how effective security measures associated with mobile ambient light sensors are
6. To propose possible solutions and risk mitigations to address any ineffective ambient light sensor security measures

1.2 Project Motivation

Sensors in general are incredibly useful in a multitude of domestic and industrial fields [see Section 2.5.2.1], this fact is especially true in the aspect of smartphones. By definition 'smart' phones are smart from their ability to interact and communicate with other external devices in many different ways. Each avenue of interaction is facilitated by some form of sensor or another. As outlined in the literature review [See Section 2.6], sensors have been extensively researched. Some types of sensors have more attention than others. E.g. there is extensive literature on smartphone accelerometers and their various phenomenon.

Due to the innate distinctive characteristics of sensors, they are able to be used in cooperation in applications such as fusing Gyroscope and Accelerometer sensors to create accurate tracking systems. [See Section 2.6] This flexibility means that they can be used to create useful systems or could be used by malicious actors.

1.2.1 Project Importance

This topic is important since there is a lack of research surrounding the effects of ALS on smartphones. ALS have the potential to be exploited and leak sensitive information about a user's movement, which in turn leads to PII violations. This project focuses on this issue to help ensure the security of ALS and prevent malicious actors from inferring sensitive information.

1.3 Project Approach

The approach that this project will take consists of several elements; a literature review covering detailed focused research and background information. The Android Java application captures the ambient light data, and finally, the data analysis process uses Jupyter Notebooks to perform signal processing and data analysis.

This dissertation will contain the necessary relevant background information surrounding computing, sensor technology and how security interacts with it all as well as a literature review on sensors, with an emphasis on ALS.

A major consideration taken is the validity of sources, by examining the credibility, relevance and reputation of academic sources. While different literature sources have been used, sources with the highest of these attributes have been prioritised. For example, sources from reputable, peer-reviewed papers have been used instead of sources from internet blogs. To search for and access these sources, research-orientated search engines or databases have been used, such as Google Scholar [2], ResearchGate [3], Scopus [4] and Royal Holloway LibrarySearch [5]

As well as presenting a literature review on ALS technologies. This paper will also present a comprehensive data analysis captured from an Android application. Through signal processing and machine learning, results are presented.

Chapter 2: Background and Related Work

2.1 Computing

The history of computing is a rich tapestry, characterised by continuous advancements and innovations that have revolutionised the way we interact with technology and subsequently, the world.

Modern computing is so important today because almost every aspect of everyday life is driven or makes use of computing to some degree [6]. One of the main reasons it permeates throughout our world comes from its inherent communication and connectivity capabilities [7]. From partaking in social media [8] to facilitating academic research, computing enables previously impossible [9] or impractical work to be carried out, such as finding the Mersenne Prime [10].

2.1.1 Computing Paradigms

As computing technologies progress, so too do the forms in which computing can take place [11]. These computing forms can be characterised by the following paradigms, Discrete fixed locations, Mobile, and Cloud computing.

2.1.1.1 Discrete fixed locations

This is the oldest form of computing [12], which simply refers to computing devices that are set in one physical location. At a fundamental level, all computing devices can be defined by this characteristic. Mainframes[13], Data centres[14], University computer labs, and Office workstations are what we consider computing in discrete fixed locations. Discrete fixed locations share the characteristics of being stationary hardware with localised processing capabilities and are self-contained in their own environment. In the case of a home desktop computer, this refers to being restrained to the physical device itself and for larger systems such as data centres, being self-contained in its physical location like a building. Due to the physical location, another characteristic is that of the encompassing environment being controlled. This means that these environments can be computationally optimised, have secure environment controls as outlined by ISO 27001 Physical and Environmental Security [15], and also have regulated internal systems such as reliable power supplies.

2.1.1.2 Mobile

The mobile technology paradigm refers to computing on portable physical devices. This became possible due to the miniaturisation of processors [16]. The smartphone has its origins in the combination of telephones and computing technology. In the 1990s and early 2000s [17] personal digital assistants (PDA) grew in popularity. Most PDAs consisted of a small physical keyboard and also had early touchscreen capabilities, however unlike contemporary mobile phones, they did not contain cellular technology [18]. As the name suggests the device was mainly for information storage and retrieval, as well as utilities through calendars and address books. As technology improved PDAs and mobile phones were merged, creating a new type of device, the Smartphone. The first being “The IBM Simon” released in 1994 [19]. As well as smartphones, which this project will focus on, mobile computing also encompasses any computational capable device that is portable, such as tablets, laptops and wearable devices. The main characteristics of this paradigm can be defined as being versatile, in that these devices can often be multi-purpose. As a result of their mobility, they are also usually battery-powered.

2.1.1.3 Cloud

Cloud services, while not strictly a unique 3rd paradigm of computing, make use of the interconnectivity of mobile and discrete devices to allow for the use of another entity's computing resources, such as data processing, data storage, and management[20]. The delivery of these services can be defined as; Software-as-a-Service (SaaS)[21], Platform-as-a-Service (PaaS)[22], and Infrastructure-as-a-Service (IaaS)[23].

2.1.2 Computing Vulnerabilities

As computing systems are ubiquitous in almost every facet of life, naturally they contain private and confidential information depending on the particular nature of that device's use.

Whether they are fixed, mobile or cloud-based computing solutions these systems will have vulnerabilities stemming from many different reasons. The most common reasons include:

- Complexity of systems: The complexity of systems can lead to an increased risk of the introduction of vulnerabilities in systems [24]. This is due to the fact that as complexity increases so too does the difficulty for developers to understand, maintain and develop code with a security-conscious approach without making mistakes
- Software design bugs and flaws: while software design flaws, bugs and errors in general do not inherently lead to security vulnerabilities, they often allow malicious actors to exploit for unintended consequences [25]. Although bugs and vulnerabilities are conceptually different, bugs relate to incorrect functionality, meaning not all bugs and flaws are security exploits but all security exploits are bugs and flaws
- Human Error: While human error includes vulnerabilities accidentally introduced in the development system of a system, human error also applies to a system's users. For example, more than 300 students in Melbourne had their health records accidentally published online by an administrator violating PII [26].
- Interconnectivity: The extensive connectivity of computing devices, especially across networks can lead to multiple attack vectors. If a device on a network becomes compromised, other devices could be at risk from an attack traversing a network. A computer worm [27] is a type of malware designed to self-replicate and infect devices on a network as an example.
- Legacy systems: By definition [28] a legacy system is a system that replies on outdated computing software or hardware. The security techniques and practises used when they were first implemented might have been performed adequately, however, as regular updating has stopped many vulnerabilities could remain exposed to be exploited by malicious attackers.
- Sophisticated attacks: complex attacks such as Advanced Persistent Threats, outlined in section 2.2.1.3 are examples of a type of complex attack that combines different techniques and methodologies to circumnavigate around a systems security defences and maintain undetected for long periods of time. For example, Stuxnet was used to attack Iranian state nuclear enrichment facilities [29].
- Insider Threats: Individuals within an organisation have the possibly to pose security risks[30], mainly from two possibilities, the individual themselves are a malicious actor or they accidentally expose the organisation to exploits. For example, an individual within a target organisation could unwittingly engage with a Phishing email [31] and devolve sensitive information.

- Insufficient security practices: Insufficient or poor implementation of security practises can lead to a system becoming exposed to vulnerabilities. For example, incorrect implementation of encryption standards and incorrect instruction on how to deal with potential cyber threats on an organisational level can leave an organisation vulnerable[32].

2.1.3 CIA Triad

In the context of computing a vulnerability can be defined as a weakness or threat in a computer system which could violate the Confidentiality, Integrity, and Availability of data, the so-called CIA triad [33] – Which is when a violation of a system's internal controls, security procedures, or its design and implementation occurs [34].

There are many different types of vulnerabilities and reasons each having complex root causes [35]. Hence, the way to respond to a particular vulnerability can depend on a variety of things. However, as the largest percentage of vulnerabilities are exploited or exacerbated by human error [36], a lot of the most effective measures are oriented around education and the implementation of data protection policies. The use and implementation of various policies can also be crucial. For example, adhering to correct comprehensive security policies to prevent the violation of CIA is crucial [36].

Other methods are also used, such as ensuring systems are regularly updated and patched to the latest secure versions. This is because as vulnerabilities are discovered they can be patched through updates disseminated by the vendor [37].

2.2 Cyber Security

The term cyber security [38] refers to the techniques and practises organisations and individuals use to reduce the risk of a vulnerability being exploited that would lead to a violation of the CIA Triad. The things that attempt to cause these violations are known as cybersecurity attacks.

2.2.1 Types of Cyber Attacks

The taxonomy of cyber-attacks can be split into two main categories [39], Active and Passive attacks. These can both be further classified into on-path and off-path attacks.

2.2.1.1 Active attacks

Involves the direct manipulation of data in a network by a hostile actor [40]. In most circumstances, this type of attack requires a direct line of connection between the target and the attacker.

Table 1 shows a selection of active attacks

Table 4: a selection of active attacks

Type of Attack	Description
Denial of Service (DoS)	DoS [41] is when a malicious entity sends a target system multiple requests, that the target system deems to be legitimate. These “legitimate” requests hold up a large portion of the system’s processing power. The target system processes these requests instead of actual legitimate requests, thus denying access to legitimate users.
Distributed Denial of Service (DDoS)	DDoS [42] is the same method as DoS, except instead of the malicious entity using one system, they use a multitude of compromised systems to perform the same attack.
SQL Injection Attack	SQL injection [43] is when malicious attackers exploit a database security vulnerability that allows users through inputting or injecting a SQL query as input data to perform unauthorised manipulation of the internal system.
Replay attack	Reply attacks [44] involve a malicious attacker capturing a communication or data and withholding it until the attacker’s desired time to resend it to gain access to or achieve an unauthorised action of a target system
Masquerading	Masquerading [45] is when a malicious attacker directly impersonates another entity in order to perform actions exclusive to the impersonated entity.
Man-in-the-Middle (MitM)	The MitM [46] attack involves an attacker intercepting communications, and placing themselves between two separate entities. Allowing interception, reading, and altering of the two entities' communications without their knowledge.

2.2.1.2 Passive attacks

Passive attacks involve the indirect disruption or manipulation of data or communication systems. Passive attacks involve indirect interaction between an attacker and the target, mainly through information leakage. Table 2 shows the different cyber-attacks that fall into passive attacks

Table 5: passive attacks

Type of Attack	Description
Eavesdropping (sniffing)	Eavesdropping [47] is the act of an attacker intercepting internal or external data communications without the target's knowledge in order to gather sensitive information.
Traffic analysis	Traffic analysis [48] is when sensitive data is analysed for behaviours and patterns that can lead to extra inferred information. This attack can also be performed on encrypted data to achieve similar pattern analysis.
Side-channel attacks	Side-channel attacks [49] involve exploiting other data streams which might not be obvious. For example, device power usage could be used to infer internal processes.
Release of message	Release of message [50] refers to the act of releasing or leaking sensitive information against the target's wishes and violating confidentiality. As an example, an attacker could intercept and disclose emails without actively manipulating the captured information.

2.2.1.3 On-path and Off-path attacks

On-path and Off-path attacks define certain modes of operation of the cyber-attacks that violate the CIA triad. These modes of operation are not mutually exclusive, in that multiple combinations of attacks could be used each under a different paradigm of attack.

On-path attackers are defined [51] as attacks where the malicious actor places themselves directly on the information channel 'path' by usually intercepting communications between the server and the client and using the intercepted information for two-way impersonation. The MiTM attack is the most well-known on-path attack.

Off-path attacks are when an attacker is not situated on the information 'path' between the server and the client [52]. If an attacker does not have access to a network the malicious actor can still 'inject' packets into a network in hopes of causing adverse effects to the system [53]. As they do not have access various techniques need to be employed, such as trying to guess the internal generation method through commonly known standards. An example would be a 'TCP Sequence Number Inference Attack' [54].

In real-world cyber-attack scenarios, depending on the target environment and situation, attackers make use of multiple different aspects of active and passive attack paradigms[55]. What type of

methods and tools attackers use depends on the target scope, for example, a multinational business enterprise would require an attack that is significantly more advanced and sophisticated compared to compromising a household network.

These sophisticated attacks are often called Advanced Persistent Threats (APTs)[56]. APTs are characterised by their prolonged, stealthy, multi-stage approach to compromise systems. Often their goal is to maintain persistent access to the target system to perform malicious activities. A famous example is that of 'Operation Aurora' in which the Chinese cyber espionage group 'Elderwood' performed a series of cyber-attacks infiltrating many American Fortune 100 companies, such as Google [57], Yahoo, and Morgan Stanley. The reason for the attack was widely believed to be to capture information on Chinese dissidents [58].

2.2.2 Cyber Attack Mitigation Strategies

The effective mitigation of cyber-attacks in general requires understanding the root methodologies of a given cyber-attack. Employing strategic frameworks are a good way to counter the threat from malicious actors against an organisation. With these frameworks in place, an organisation can effectively mitigate the risk[59].

2.2.2.1 Cyber Kill Chain

There are many ways in which a cyber-attack can be mitigated or prevented and while they follow a general layout [60] Figure 1 shows the industry standard Lockheed Martin's Cyber Kill Chain (CKC) model [61], a phased-based methodology outlining and interpreting the typical structure and sequence of events of a cyber-attack. This methodology was created to allow organisations to better equip themselves to deal with cyber threats. If an organisation can understand each step, it can focus its defence on organising mitigation or defence strategies for each phase of the model.

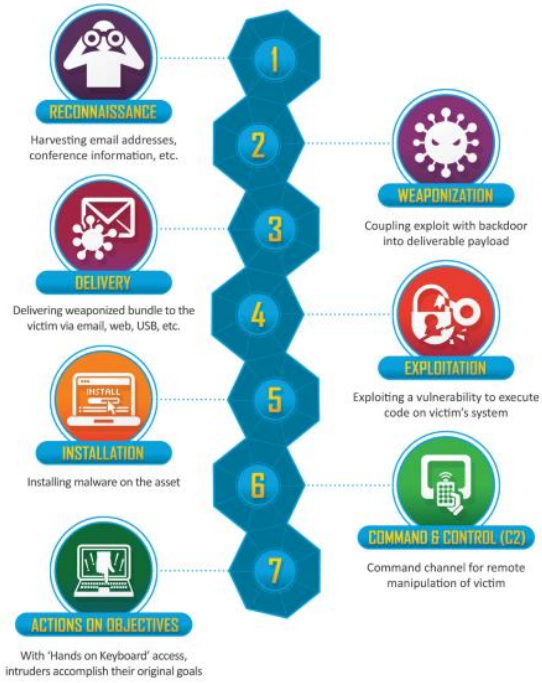

Figure 1: Lockheed Martin CKC model - [61]

As outlined in Figure 1, the Lockheed Martin CKC Model uses seven stages for a stereotypical cyber-attack. These stages are outlined below:

1. **Reconnaissance:** in this step, an attacker attempts to collect as much information as possible surrounding a target in order to understand the scope and the requirements needed for the adversary to penetrate the target. Many active or passive approaches can be taken, for example, social media information can be taken passively to understand the hierarchy of the organisation. Active target enumeration tools can be performed to scan networks for open ports or vulnerable services such as NMAP [62].
2. **Weaponisation:** in this step, the malicious actor designs their attack approach using the information from the reconnaissance stage. Knowing the layout and what services and ports are vulnerable the attacker can choose their approach and which element of the target organisation to exploit. For example, social engineering phishing attacks [31] can be used to exploit the vulnerable human element. Malicious actors could also network attacks to exploit vulnerable open ports or services.
3. **Delivery:** After the attack has been weaponised the malicious actors have to deliver the payload in some way. Depending on the target organisation, different ways exist, such as malicious downloads, physically dropped USBs or in which an employee or target unknowingly plugs into the target system.
4. **Exploitation:** Once the attacker has successfully delivered their payload to their intended target then the actual mechanism is engaged to exploit the chosen vulnerability to achieve the malicious actor's goals.
5. **Installation:** In this stage, the attacker has gained access to the system and exploited the vulnerability. Not all types of attacks follow this step and propagate through a system. However, if applicable the malicious actor in this step would gain access to other devices on the network through remote access trojans (RATS) [63]. Once a RAT or similar malware is present the malicious actors can implement backdoor access for maintaining a connection to the target
6. **Command and Control (C2):** After an attacker has established a presence on a target's system then using the chosen exploitation and installation methods the malicious actor can further command and control the system to achieve the malicious actor's aims.
7. **Actions on Objectives:** This stage is the actual enactment of the malicious actor's aims and goals through the C2 establishment. For example, a RAT could be installed on a smartphone and data exfiltrated from the unaware user.

Due to the ever-changing landscape, no two cyber-attacks are truly the same. Because of this, having an all-encompassing cyber threat methodology is nearly impossible. However, making use of multiple frameworks helps counter this and aids in creating a more complete image. Whilst the CKC model is the most common paradigm, the MITRE ATT&CK Framework [64] and NIST Cybersecurity Framework [65] also exist to compensate for the CKC's weaknesses. The CKC's pros and cons [66] are listed below in sections 2.2.2.1.1 and 2.2.2.1.2:

2.2.2.1.1 Pros

- The CKC model emphasizes preventative measures by outlining the early steps of a cyber-attack through the inclusion of the reconnaissance stage. Meaning that organisations are able to reduce the risk of cyber-attacks by lowering their exposure to outside threats

- The CKC model follows a clearly structured approach as outlined in Section 2.2.2.1, allowing for organisations to assess each step individually and apply the corresponding mitigation strategies.

- As one of the first comprehensive security mitigation frameworks, the CKC model is widely used by many organisations.

2.2.2.1.2 Cons

- The CKCs model is based on an older military kill chain [67] which has seen no adaption for a long period of time, which means that some of the emergence of cyber threats might not be taken into account.
- While the model outlines each phase of an attack it does not directly suggest a solution to counteract against each phase.
- The CKC model does not take into account other types of attack well, while external threats are modelled, internal attacks are not as well acknowledged.
- The CKC model does not take into account iterative attacks, or attacks combining multiple adaptive malicious tactics, techniques and procedures.
- While the CKC model outlines an attack from beginning to end, focused on their actions executions and goals it follows a rigid structure which makes the approach less successful for more fluid attacks.

2.2.2.2 MITRE ATT&CK Framework

The Adversarial Tactics, Techniques and Common Knowledge (ATT&CK) framework produced and maintained by MITRE provides an extensive and constantly updated description of actions that attacks take. The high-level goals, which are the tactics and the specifics covered in the technique category. This framework provides analysts with specifics when dealing with known adversaries. This can be incorporated into the CKC model to gain further insights into each step on the kill chain, as well as potential gaps in an organisation's defence.

Figure 2 shows the ATT&CK Frame work that addresses some of the weaknesses of the CKC model. By providing a knowledge base of known adversarial tactics and techniques based on real-world observations this improves on the CKC, which uses an older military kill chain. The ATT&CK framework also improves on the lack of direct solutions in the CKC model, through providing detailed techniques for each phase.

Figure 2 also shows active mitigation strategies that organisations can employ to defend against a cyber-attack, it is also of note that the frame work has more phases to account for the differing potential variability in cyber-attacks. The framework's pros and cons are listed below in sections 2.2.2.2.1 and 2.2.2.2.2:

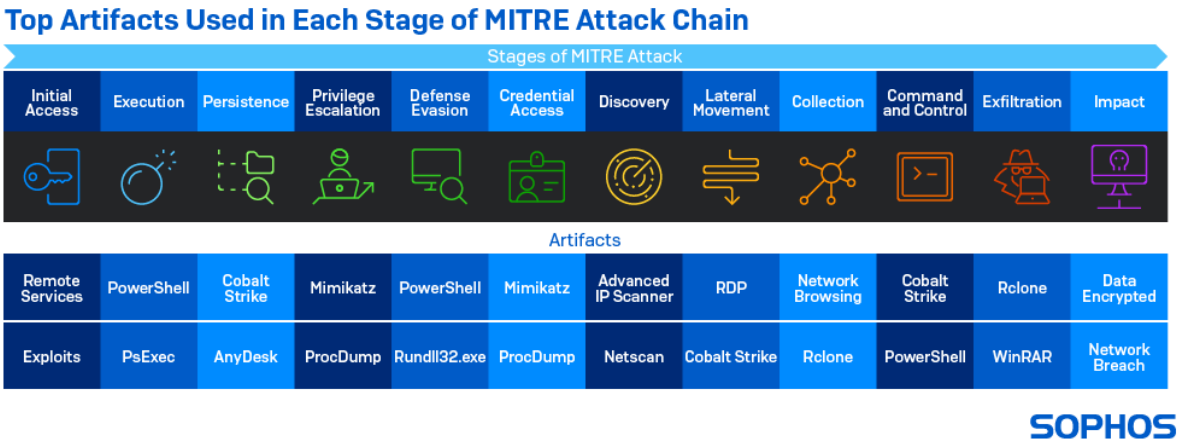

Figure 2: MITRE ATT&CK Framework – [68]

2.2.2.2.1 Pros

- The ATT&CK Framework offers cyber security professionals specific details as to how malicious actors operate. In turn, allowing for defence strategies to be implemented.
- As the framework is centred around real-world data, the tactics and techniques outlined are up-to-date and relevant. This allows for informed security decision-making on an organisational level.
- Many Security Information and Event Management Tools (SIEM) such as Securonix [69] integrate the framework allowing organisations to automatically apply the framework to their systems.

2.2.2.2.2 Cons

- The level of detail within the framework can be overwhelming with the amount of information displayed. This is especially true for organisations with limited cyber security experience.

- Due to the complexity and level of detail, the number of resources an organisation requires to enact the framework can be intensive. Often requiring specialised personal to apply the tactics and techniques.

- While the framework is comprehensive and based on real-world data, this framework is unable to make predictions and react to new emerging threats such as 0-day exploits [70].

2.2.2.3 NIST Cyber Security Framework

The NIST Cyber Security Framework (CSF) is designed to help businesses and organisations manage, understand and reduce their cybersecurity risks. It was designed by the National Institute of Standards and Technology (NIST) [71] Unlike the aforementioned frameworks, the CSF outlines the best practises that an organisation should take should it choose to do so in general.

Figure 3 shows the NIST CSF framework, which consists of Identify, Protect, Detect, Respond, Recover, and Govern. Each aspect covers a fundamental part of an organisation's cybersecurity posture. The CSF takes a holistic approach, meaning this approach is designed to fit any type of organisation. The framework's pros and cons are listed below:

2.2.2.3.1 Pros

- The CSF provides a flexible approach that any organisation can adapt to fit their exact needs and requirements.
- This framework is widely acknowledged and comes from NIST a reputable source. Therefore, it has seen wide spread adoption in the industry. Meaning that different organisations can use the foundations laid out in this framework to facilitate security-based collaboration.

2.2.2.3.2 Cons

- As the CSF is designed to provide high-level guidance, it lacks detailed and actionable information.
- The broad scope the CSF provides can make it challenging for organisations with a lack of cyber security expertise to properly adapt and action the framework.
- The CSF is a voluntary framework, meaning that no third-party organisation requires it for compliance. This can lead to some organisations failing to implement it correctly, or entirely.

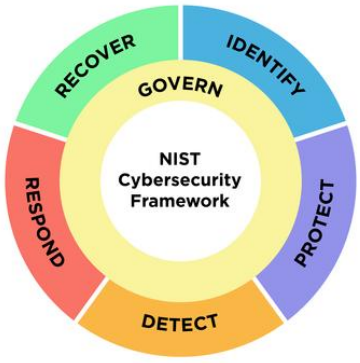

Figure 3: NIST Cybersecurity Framework - [65]

2.3 Mobile computing

Mobile computing [72] refers to computing devices that are not fixed to one location and are usually categorised by being battery-powered. This section further outlines the paradigm of mobile computing.

2.3.1 Mobile Device types

There are three main classes of computing devices that satisfy the aforementioned attributes. These classes are; Mobile phones [73], Portable Computers [74], and Wearable Computers [75].

2.3.1.1 Mobile phones

Mobile phones can be considered one of the most ubiquitous classes of mobile computing devices [76]. Originally mobile phones were primarily used for voice telecommunications, but with the fusion of technological advancements, these devices are now used for a wide range of applications and services [77]. Examples of devices that fall under this class include, basic feature phones and smartphones running various operating systems such as Android or iOS.

2.3.1.2 Portable Computers

Portable computers are devices that, while they are still portable, are larger than other mobile phones and smartphones. Laptops, tablets, and notebooks fall into this category. Portable computers trade an aspect of their portability for increased versatility through larger hardware components. For example, while a laptop is portable it cannot be as easily transported compared to a smartphone.

2.3.1.3 Wearable Computers

Wearable computers are computing devices designed specifically to be worn on the human body to sense, compute, run applications and allow connectivity to other devices and services. Examples include smartwatches, smart glasses and fitness trackers.

Table 3 shows the various paradigms of mobile computing devices and their associated attributes.

Table 6: Comparison of mobile device types and their attributes

Mobile Device types	Description	Battery Powered	Portability	Connectivity	Versatility	Personalisation
Mobile Phones	Originally an evolution of landline telecommunications, mobile devices now include a range of apps and services.	Yes	Designed to be able to carried or operated in a user's hand	Generally, supports WIFI, Bluetooth, and 4/5G. Extra features vary by model	Focused on lightweight applications like productivity tools and games	Basic customisation of settings and apps, often limited by the manufacturer.
Portable Computers	These are devices with more power than mobile phones and utilise similar architecture as conventual discrete fixed locational computing	Yes	Less portable than mobile computing, but more portable than discrete fixed computing	More emphasis on port connections, such as USB, Ethernet and HDMI	Similar functionality to discrete fixed computing such as a desktop	Full technical personalisation similar to desktop devices
Wearable Computers	Devices that are focused on sensors and designed to be worn by a user, for example, fitness trackers and smartwatches	Yes	Very portable, designed to be always situated on a user's physical body.	Contains limited ports and connects via Bluetooth or WIFI	Similar to mobile phones, but has an emphasis on specialised fitness applications	A similar level of customisation as mobile phones.

2.4 Android Smartphones

To understand the security model employed by Android, the ecosystem background is crucial. One of the main reasons why the Android operating system is used widely around the world is due to its open architecture, which in turn allows for many different original equipment manufacturers (OEMs) [78] to utilise the Android stack for creating new related, but independent devices [79]. At a basic level, the Android Open-Source Project (AOSP) [80] provides OEMs with a basis to further develop. While the usage of the Android stack is decentralised [81], there is a caveat of compatibility that OEMs need to follow. To which extent OEMs utilise the stack governs which standard must be adhered to. For example, at a basic level, the Compatibility Test Suit (CTS) [82] criteria must be achieved. At higher levels involving more basic components, more standards must be followed.

2.4.1 Android Threat Model and Ecosystem

While similar to standard other operating systems, the Android threat model's attack surface is slightly different [83]. For example, unique to mobile devices, attacks can be carried out on physical access controls [15], by performing NFC relay attacks [84] which can be used for gaining unauthorised access to contact payment systems, and also for gaining unauthorised access control as these operate on the same system.

Recent works have described the differing threats to Android mobile computing and traditional computing. Mayrhofer [85], states that these attacks are summarised under four categories, Physical, Communication, Platform, and User interaction.

- **Physical attacks** consist of loss or theft of a device, leading to unauthorised usage.
- **Communication attacks** consist of subverted communication in a guise to gain unauthorised access to a system.
- **Platform attacks**, utilising the vulnerabilities innate in Android components such as abusing APIs and utilising platform exploits to attack other internal applications.
- **User interaction attacks** involve tricking the user via masquerading attacks to capture user inputs.

While there is an overlap with the issues that traditional discrete fixed location computing and mobile computing, some issues are unique.

2.4.2 Android Platform Security Model

Taking into account the ecosystem context and threat model, the Android security model emphasises a balance between the privacy and security requirements of users with the needs of the applications and services present on the platform itself. As stated by Mayrhofer [86] the Android Platform Security Model is defined by 5 rules:

1. **Multi-party consent:** This states that no action should be performed without the consent of all parties involved. The parties usually consist of the user, the developer, and the platform and if one party vetoes a decision the action is not carried out. This idea extends between subjects (applications and user processes) and objects (files, network sockets, memory regions, and other underlying constituent parts). However generally if an entity creates data or an object the creator controls it and has privileged control over it. While this is true the location of data also matters, for example, data in the shared storage location is accessible via all users and data in private app storage directories are governed by their creator applications.

2. **Open ecosystem access:** This states that inter-app communication is explicitly supported on the platform and central vetting and registration are not required of developers and users so much that they follow the basic tenets of the AOSP.
3. **Security as a compatibility requirement:** This states that devices must adhere to the Compatibility Definition Document (CDD) [87] which acts as a 'hub' for reference and states that new devices must be compatible with new Android versions and their security benefits. Devices that do not adhere to the CDD and CTS are not considered Android devices.
4. **Factory reset restores the device to a safe state:** This states that any Android device must be able to be returned to a state where only integrity-protected applications and systems are present and do not require essential software to be reinstalled in the event of a persistent cyber security breach.
5. **Applications are Security principles:** This states that applications should not be run or considered as fully privileged agents of the platform. Meaning that having control over a user's application would not automatically grant a malicious actor root privilege. Instead, applications act as individual sandboxed environments with limited interaction with other applications on the same level. It also means that applications do not have permission to make functional changes on systems that affect other applications and services on the device.

These foundational rules are applied to the Android stack (see Figure 4)

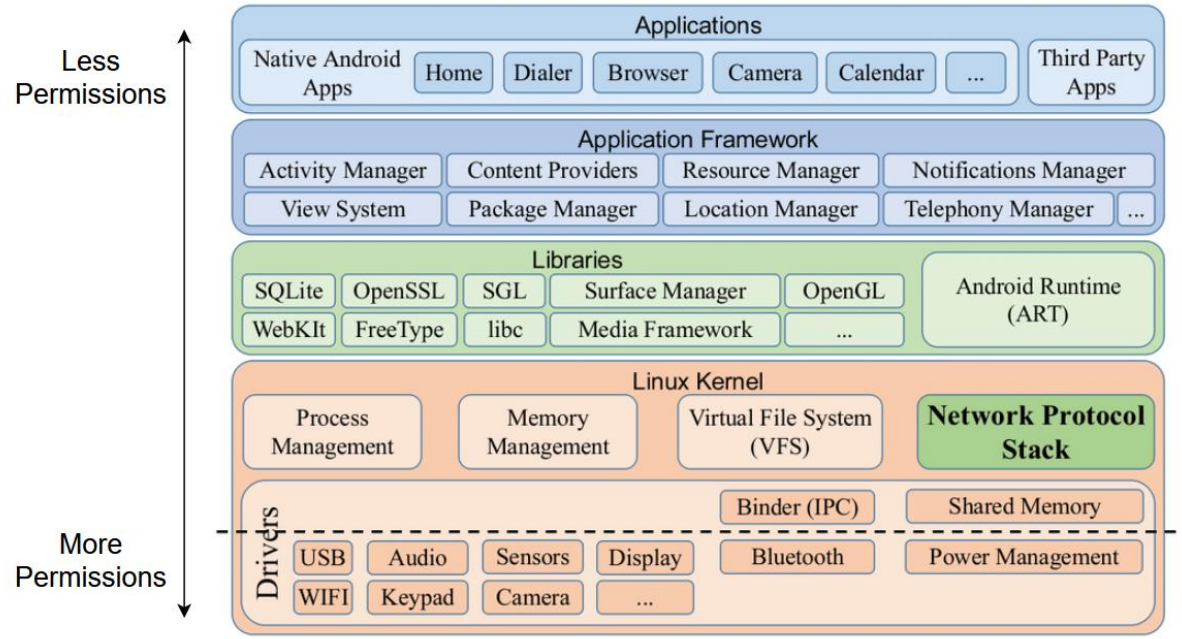

Figure 4: Android software stack - [88]

2.4.3 Android Platform Attack Vectors and Mitigation

It is crucial to understand how cyber-attacks target the various aspects of the Android stack as shown in Figure 4. The following section outlines some of the threats and the appropriate mitigations each layer of the Android stack faces.

2.4.3.1 Application layer

The application layer consists of both standard native Android applications and third-party applications. This is the layer that the user has the most direct interaction with, and so where the majority of user-based functions are performed such as permission consent.

2.4.3.1.1 Threats

The versatility of Android often means that applications end up handling sensitive user data, such as PII [89]. Therefore, one of the common threats present on the application layer is that of data leakage. A user's data could unintentionally be accessed by a malicious actor. This could be caused by many things such as, data protection procedures being followed incorrectly, inadequate permission management and weak authentication and authorization systems [90].

2.4.3.1.2 Mitigation

The mitigation strategy for handling potential data leakage is to ensure proper application sandboxing [91], having applications run in isolation helps ensure data protection and following the idea of multi-party consent to ensure that an application cannot access PII without the user's permission, therefore reducing the risk.

2.4.3.2 Application Framework layer

The application framework layer hosts the essential APIs and services that constitute applications [92]. This layer hosts inter-app communication services as well as critical application management systems such as the notifications manager, allowing applications to display alerts such as received messages [93].

2.4.3.2.1 Threats

The application framework layer employs inter-application communication (IAC) mechanisms, such as Android's intent system [94], which allows the launch of other applications through the use of startActivity. Vulnerabilities in IAC mechanisms exist and have been exploited for financial gain. Holberg [95] demonstrated Android Intents against Swish [96] in iOS and Android operating systems for financial gain.

2.4.3.2.2 Mitigation

Some of the mitigation strategies for protecting against IAC attacks are through the use of regular updates to patch potential vulnerabilities and keep IAC API services secure by ensuring that proper procedures are followed [97]. For example, guaranteeing that data incoming to the IAC mechanism is authenticated and validated to minimise the risk of a malicious action. Multi-party consent is also crucial, especially when PII or financial information is involved.

2.4.3.3 Libraries layer

The libraries layer of the model provides the various code bases that applications rely on, also including libraries for graphics, data handling and communication as well as other management services and the Android Run Time (ART) [98] ART primarily performs ahead-of-time (AOT) compilation of an application into native machine code for optimisation and security.

2.4.3.3.1 Threats

The primary threat to the libraries layer is through the improper usage of libraries, either through not regularly updating libraries or allowing unverified third-party libraries which may include malicious content. For example, the Mavensgate attack [99] exploited poorly maintained or abandoned libraries in ApacheMaven [100] repositories via malicious attackers enforcing DNS TXT record [101] rights to allow for malicious library updates to be pushed and distributed as legitimate packages.

2.4.3.3.2 Mitigation

The mitigation strategies for reducing the risk that threats to the libraries layer pose, involve ensuring that libraries are regularly vetted and authenticated, as well as making sure that secure integration practises are followed. For example, app developers should not implement depreciated libraries or functions [102]. Android's applications are security principles, meaning that infected applications are sandboxed and isolated away from other processes and applications, reducing the impact of a compromised library [91].

2.4.3.4 Linux Kernal layer

The Linux kernel layer is the foundation for the Android operating system [81]. It is responsible for managing hardware resources, such as handling device drivers and controlling inputs from peripherals such as USB ports and sensors. It is also responsible for providing core system programmes and services and enforcing the strict access control sandboxing rules as well as permissions for the subsequent functions and applications. While this dissertation refers to the kernel as the 'Linux Kernal', the Android Kernal is an adaption of the Linux Long Term Supported (LTS) kernel [103]. LTS kernels are combined with updated patches to create the kernel layer.

2.4.3.4.1 Threats

Due to the importance of the Kernal layer, the impact of possible threats can be crucial. If a malicious actor compromises the Kernal layer, then it can lead to major consequences [104]. For example, malicious actors could gain root access to a device leading to system-wide security compromises. It could also allow malicious actors to install hard-to-detect malware such as RATS to provide persistent access to the attacker's server, potentially violating the security of the entire system, and opening up the possibility for attackers to take over the device and perform lateral network attacks. DirtyCOW [105] is a race condition vulnerability in the Kernal layer that affects the subsystems involved in the Copy-on-Write mechanism. The exploit allows unprivileged users to gain write privileges on read-only memory locations. Meaning malicious actors could inject arbitrary code.

2.4.3.4.2 Mitigation

To mitigate the impactful threats posed against the Kernal Layer, a defence-in-depth approach is needed. Kernal hardening techniques such as the Kernal self-protection Project (KSPP) [106] introduce security features such as Address Space Layout Randomisation (ASLR) [107] protecting against buffer overflow attacks by preventing malicious actors from working out external application address space contents through randomisation. SELinux [108] can also be used for Kernal hardening. This system enforces mandatory access control (MAC) [109] policies, which govern what processes and applications can do even when they are compromised in turn protecting sensitive information against unauthorised access via compromised elements.

Regular updates and patching of the kernel can protect against exploits [110], for example, more recent android kernel updates address the issues caused by the DirtyCOW exploit. The factory reset mechanism as described in section 2.4.2, provides another mitigation strategy. If a system is compromised at a fundamental level, then acting as a last resort a factory reset will wipe all device data and reinstate the original trusted kernel configuration. In turn, removing the majority of persistent malware on a device.

2.4.3.5 Hardware Layer

Below the android software stack sits the hardware layer, while not directly related to Android, the Hardware layer consists of independent physical elements that the kernel interacts with via drivers. For example, the hardware layer consists of processing units such as the CPU and GPU. Memory and storge devices containing persistent and temporary data storage. Input/output devices, such as keyboards, speakers, display screens and other peripherals. The device sensors are also present on the hardware layer. The Android system that controls the interaction between hardware and the android stack is also known as the hardware abstraction layer (HAL) [111], as this layer allows developers to interact with lower-level hardware through the abstraction of hardware specifics.

2.5 Mobile Sensors

This section will expand on the sensors described in Section 2.4.3.5.

2.5.1 Common Sensors

While there are some common shared hardware sensors across mobile devices, such as Accelerometers and Proximity sensors [112], there also exist less used sensors. This is partly due to the decentralised nature of the android ecosystem. OEMs take into consideration the type of device being created and its use case [113]. For example, Wearable computing devices will have an emphasis on fitness sensors, meaning there is an increased chance for those devices to contain heartrate or oxygen sensor technology.

Table 4 below shows the common sensor employed in the android ecosystem and their applications and functionalities. It is also of note that these sensor technologies are not unique to the mobile computing paradigm.

Table 4: Sensor types, their applications and functionality

Sensor Type	**Applications and Functionality**
Ambient Light	The ambient light sensor [1] is used to measure the intensity of light in a phone's environment. Used mainly to adjust the display's brightness for optimal power efficiency and visibility.
Camera	The camera [114] is used to take photographic images and videos. Quality varies based on camera resolution
Microphone	The microphone [115] is used to detect sound waves and convert them into electrical signals. Can be used in voice calls and voice commands
Proximity Sensor	The proximity sensor [116] uses an infrared LED and an infrared light sensor to calculate the distance from an object. Mostly used to turn the screen off during a phone call.
Magnetometer (compass)	The magnetometer [117] is used to detect magnetic fields and provide navigational orientation for mapping applications.
Global Positioning System (GPS)	The GPS [118] interacts with satellites in orbit to accurately determine geographical position. Used in mapping and navigation software.
Light Detection and Ranging (LiDAR)	The LiDAR [119] sensor is used to measure depth and distances via emitted laser pulses. Used to create 3D maps of objects
Accelerometer	The accelerometer [120] is used to measure vibration, tilt and acceleration along the x, y, and z-axis. Mainly used to determine screen orientation.
Gyroscope	The gyroscope sensor [116] is used to provide more accurate angular velocity data than the accelerometer. Used for video and image stability.
Fingerprint	The fingerprint sensor [121] is a biometric authentication system so that only the approved users can gain access to the device.

Sensor Type	Applications and Functionality
Touchscreen	The touchscreen sensor [122] detects localised pressure on the screen surface via changes in electrical currents. Used to allow users to interact with the touchscreen.
Barometer	The barometer sensor [123] is used to measure atmospheric pressure. Used to measure weather patterns and estimate altitude.
Pedometer	The pedometer [124] is used to count the steps of the user, determined by values outputted by the accelerometer. Used to track physical fitness.
Thermometer	The thermometer sensor [125] comes in two types. Internal thermometers are used to control device temperature to prevent overheating and external thermometers that measure outside ambient temperature.
Heart Rate	The heart rate sensor [126] is used to track a user's heart rate via LED and optical sensors. Used in health applications and measuring stress levels.
Air Humidity	The air humidity [125] sensor is used to measure the ambient air humidity.
Hall	The hall sensor [127] is used to measure and detect changes in magnetic fields, usually from devices. Used to detect magnets in flip covers or devices like VR goggles.
Geiger counter	The Geiger counter sensor [128] is used to measure the amount of ionising radiation in the environment. While not a common sensor its existence demonstrates the versatility of smartphone sensors.

2.5.2 Ambient Light Sensor (ALS)

This section will outline and highlight the ambient light sensor touched on in table 4.

2.5.2.1 Ambient Light Sensor Usage

Ambient light sensors [1] are hardware sensors that use various methods to detect and measure received light, in units of lux. Lux is the measured light over one square metre [129].

ALS components are an essential part of many different industries, due to their ability to measure the surrounding ambient light levels. The ALS market cap is predicted to reach “$15.7 billion” as of 2030 [130] The most common form of ALS technology is in consumer electronics, which this paper focuses on. The following section highlights the various industries that benefit from ALS technologies with their functionality and purpose.

2.5.2.1.1 Consumer Electronics industry

ALS technology in the consumer electronics industry has widespread adoption in handheld devices [131], such as smartphones and laptops. In which the automatic adjustment of screen brightness is based on the ambient light conditions. The main benefit is reducing battery usage by improving the energy usage efficiency of the device. The main purpose of this is to provide the best optimal screen visibility for the user whilst also conserving the device's battery life.

2.5.2.1.2 Automotive industry

ALS technology is integrated into a large portion of adaptive internal and external automotive lighting systems. External lighting systems allow for headlights to automatically switch between high and low beams in accordance with the ALS conditions. This increases the safety of headlights as the correct luminescence allows the driver to see without blinding other drivers, in doing so increasing overall road safety for everyone. Internal lighting systems also exist, these adjust the light inside the car and its dashboard to be comfortable for the human eye [132].

2.5.2.1.3 Smart home industry

ALS technology is also employed in the smart home industry through quality-of-life devices, such as automated windows, blinds, and smart light bulbs [133]. These devices work via, measuring the available natural light levels and acting accordingly and create a comfortable environment according to the user's prerequisites. As with most ALS industries the main purpose is to lower energy usage and increase user comfort.

2.5.2.1.4 Medical Industry

ALS technology is also used in the medical and healthcare industry. ALS has seen some use in improving the tracking of sleep schedules and monitoring patients' natural light exposure [134]. Its main purpose is to aid in providing accurate information for a patient's medical professional to help make decisions to help.

2.5.2.1.5 Industrial Automation

ALS technology can also be used in factories and other manufacturing situations [135], in a lighting capacity. In an effort to reduce costs, some industrial organisations use ALS to adapt lighting systems to save on costs whilst also maintaining the required light levels for optimal productivity and safety.

2.5.2.2 Different types of ALS Technologies

As seen in the previous section there are many use cases for ALS, each industry having different requirements leading to different ALS solutions. Each ALS type has its own construction with advantages and disadvantages. Figure 5 below highlights what the section below will describe with an emphasis on photodiodes and phototransistors.

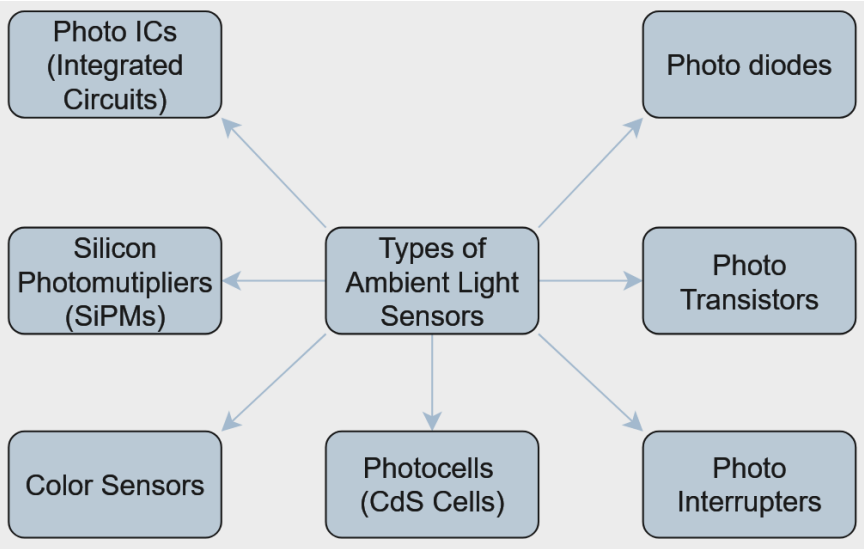

Figure 5: Types of Ambient Light Sensors

2.5.2.2.1 Photo Diodes

Figure 6 shows a Photo Diode, these are semiconductor devices that are able to generate a voltage or current when there is light exposure. This is done by photons with sufficient energy colliding with atoms on the substrate material, releasing an electron and creating an electron-hole pair via the Photoelectric effect [136]. These pairs then create the flow of electric current. The flow being determined by the number of photons colliding with the sensor, the subsequent current being directly proportional to the light intensity. Photo diodes operate in two modes: photoconductive and photovoltaic modes. Photo conductive generates a current whereas photovoltaic generates a voltage.

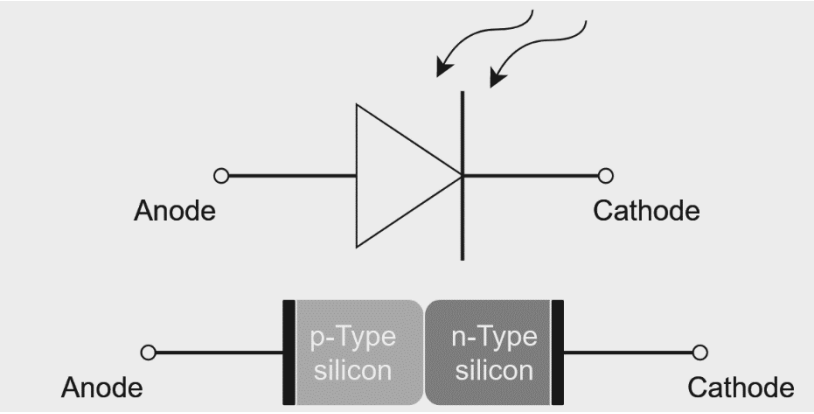

Figure 6: depiction of a photodiode

2.5.2.2.2 Phototransistors

Figure 7 shows a Phototransistor, these are devices that combine the different properties of transistors and photodiodes [137]. They are designed to convert light input into electrical current, allowing for increased control and amplification of the output signal. Phototransistors are constructed of three semiconductor layers; the base, emitter and collector, these layers form the P-N junctions; different configurations also exist such as P-N-P and N-P-N. It is also important to note that configurations without the base also exist as the base allows for a threshold to be set. When light falls onto the base-emitter region electron-hole pairs are formed, these are then allowed to flow across either the emitter or collector as current. The small current created by the base-emitter controls the much larger current flowing from the collector to the emitter creating gain or amplification.

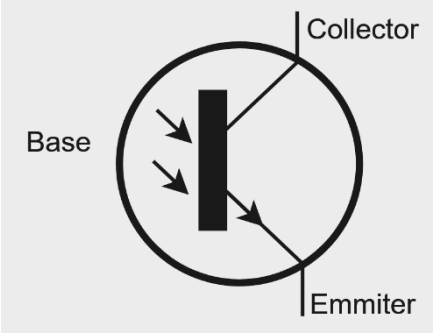

Figure 7: depiction of a photo transistor

2.5.2.2.3 Other Light Sensors

Photodiodes and photo transistors are the most common forms of ambient light sensor technology employed in smartphone systems [138]. This is because their simplicity makes them cheap to manufacture while maintaining a high level of sensitivity and fast reaction times.

Other common light sensor technologies include; Photo Interrupters [139] [see Figure 48], Photo CdS Cells [140] [see Figure 49], Colour Sensors [141] [see Figure 50] and Silicon Photomultipliers [142] [see Figure 51]. The way that these ALS technologies are integrated into smartphone technology also affects their use. For example, there are two common integrated circuit types [143]; Digital [see Figure 52] and Analogue [see Figure 53].

2.6 Related Works

This section will discuss the existing literature surrounding the attack surface of android devices, enacted through vulnerabilities and characteristics present within said system, with a focus on smartphone sensors and the role the ambient light sensor plays in information leakage.

This section will cover relevant literature on the security vulnerabilities in android systems, motion sensors, device fingerprinting, sensor fusion, and light-based attacks.

Partly due to the frequent everyday use of the android stack, there exists a plethora of security vulnerabilities. As described by Bhat et al. [144] These can include Hardware-based attacks[144], Kernel-based attacks[144], Hardware abstraction layer-based attacks[144], and application-based attacks[144]. Aalqazzaz et al. [145] also describe indirect methods, such as, power side-channel attacks[145], public resources side-channel attacks[145], and Motion sensor side-channel attacks[145]. While these categories are true representations of attacks on the Android stack, this dissertation focuses on hardware-based attacks.

There also exists a plethora of security mitigations against these attacks. For example, in work done by Bhat et al. [144], mitigation tools are outlined, such as Apex [146] which provides extensions to the Android permission model allowing for restrictions to resource usage and granting selective permissions. XManDroid [147], a security framework for extending the monitoring mechanism for detecting a preventing escalation attack. It can be noted that due to the complexity of the Android platform, with each attack targeting specific technical abnormalities having universal mitigation strategies becomes increasingly difficult.

Side channel-based attacks while not based on direct attacks against innate technical vulnerabilities in the android stack but are instead based on indirect means of attack such as information leakage, these still have the same specificity issues with mitigation strategies due to complexity.

While the aforementioned techniques such as kernel hardening can reduce the attack surface area, this paradigm relies on information leakage. For example, Han et al. [148] demonstrated PII violation by using accessible accelerometer data to infer the starting point and trajectory of an individual.

2.6.1 Motion sensors

This section will focus on side channel motion sensor attacks and the related works.

In general motion sensors constitute two different sub-paradigms [149], hardware-based and software-based. The accelerometer, gyroscope and to a lesser degree proximity sensors are hardware-based, and built using these as a base, the software sensors include gravity, game rotation vector, and linear acceleration. These motion sensors whilst providing useful motion data have also been used to infer PII, such as location data [148], [150].

Another example is that it has been proven that a user's keypad inputs can be secretly inferred via the impact of their taps on the device, this was shown by Xu et al. [151]. Further work by Maiti et al. [152] has used this technique to capture PIN entry data on a mobile device. Michalevsky et al. [153] have also used similar motion sensor attack vectors to achieve a success rate of 50 per cent when classifying speaker identity using gyroscope motion sensors. Implemented with a limited dictionary success rates rose to 65 percent. Zhang et al. [154] also used the suite of motion sensors for hot word detection. The sound recording capabilities of gyroscopes are limited, only being able to pick up signals lower than 200 Hz. [155] Human speech is around 116 Hz to 217 Hz. As a result, this opens up the possibility for potential eavesdropping attack vectors especially if sample rates are improved, thereby increasing the classification rates.

As mobile devices are by definition mobile, this means that just by having a device present on a user's person they are producing information that could be used to classify their motion type. For example, if the user is in a car, on foot or running. Bedogni et al.[156], among other researchers [157], [158], have demonstrated that motion sensors can infer these motion types. These works have been carried out by performing feature extraction and data classification modelling on gyroscope and accelerometer inferred data. Other researchers [159] also performed similar classifications using the magnetometer and linear accelerometer. Zhang et al. [160], showed that motion sensors have a legitimate energy-efficient location tracking use case, they used the orientation sensor, the accelerometer along with Wi-Fi as a substitute for the power-intensive Global Positioning System (GPS).

While attacks on a user's movement have been shown possible, researchers have exposed that motion sensors have also been shown to have a use case in handwriting authentication related to wrist movement [161], [162], [163]. Further work has shown that accurate motion predictions can be performed. Maiti et al., [164] also showed that the motion sensors on a wrist wearable device can be used to infer the combinations to mechanical locks as motion and rotational information is leaked as a user unlocks a lock via wrist rotation. Mylonas et al., [165] also expressed the credibility and admissibility of sensor data, such as motion sensors in a court of law and discussed the inherent characteristics of smartphone data and their use to law enforcement agencies. Proving the high standard and admissibility of inferred motion sensor data.

2.6.1.1 Device Fingerprinting

While information leakage from motion sensors can be susceptible to attacks, another attack vector is that of device fingerprinting. As sensors are physical objects, needing to be fabricated and assembled to take very precise measurements, they are susceptible to negligible manufacturing irregularities. While the irregularities would be unaffected and uniform in their output, it is possible for two sensors to give different average readings. The difference being the manufactured irregularities, which can be taken to uniquely identify a sensor. Das et al. [166] showed this to be true by utilising the discrepancies found in gyroscope and accelerometer outputs to allow internet entities to track users in lieu of browser cookies as some Android sensors do not require express user permission to be accessed [167].

Thereby circumnavigating the need for the user's explicit permissions. Dey et al. [168] demonstrated this phenomenon on 25 Android phones and 2 tablets with a 96% accuracy, whilst taking into account real-world conditions. Bojinov et al. [169] further show permission avoidance via the manufacturer's calibration errors in the accelerometer sensor. This time being accessed through JavaScript running in a mobile browser. While not permission-adverse, they showed the same tracking ability through similar discrepancies in the microphone-speaker system.

2.6.2 Sensor fusion

The feasibility and accuracy of user tracking and environment fingerprinting from information leakage sources increase the more sensors are used. This is shown through SurroundSense, Azizyan et al. [170] showed that the combination of ambient light, colour, the sound captured by a phone's camera and built-in microphone as well as the movement gathered by the accelerometer allows for an accurate logical localization rate of 87 per cent. Shen et al. [171] showed the combination of an accelerometer, gyroscope and magnetometer can be used to track user input on their smartphone device. Malti et al. [172], Wang et al. [173], and Lui et al. [174] have also explored motion/sensor-based inference of user keystrokes on smartwatches. Miluzzo et al. [175], Cai et al. [176] and Owusu et al. [177] show that similar techniques of using the accelerometer and gyroscope can leave smartphones vulnerable too. These related works highlight the security and privacy issues that

multiple compromised sensors present. Although some sensors require express permission from the user to access, they nevertheless are still a valid attack vector as often users misinterpret perceived risk as actual risk, as shown by Mehrnezhad et al. [178] thereby allowing access to sensors that could open up potential attack vectors. Simon et al. [179] also showed that when granted access to the microphone and camera, a device's pin can be inferred. The camera is used to track device orientation and the microphone is used to infer tap events. Further showing the applicability of sensor fusion side-channel attacks.

2.6.3 Light-based attacks

Although the movement sensors, such as the accelerometer are one of the most widely discussed side-channel attack vectors in the literature. Do et al. [180] presented over 100 research papers just on visible light communication (VLC) systems utilising light as an information transfer vector. As light offers another valuable side-channel attack vector. Backes et al. [181], [182] showed that information leakage is possible. Through the reflections of diffused light off objects, such as the user's eyeball, shirt and glasses. Allowing for the original, possibly confidential data displayed on the source screen to be reconstructed. Kuhn [183], showed that emanations from a cathode ray tube (CRT) display can be reconstructed from the diffused reflection on a wall and showed that in low-light levels, the reconstruction is very effective. He used a “Hamamatsu H6780-01 photosensor module” which contained a photomultiplier tube (PMT). Later experiments by Schwittmann et al., [184], [185] achieved the same outcome using ambient light sensors. Xu et al., [186] demonstrated a similar experiment from a distance of 70 metres. Chen et al. [187], and Sun et al. [188] expanded on earlier work, they demonstrated a keystroke inference attack via the analysis of a user’s eye movements and also showed that recording the motion of a device can be used to infer a user’s typed inputs. All culminating is further violations of PII information.

Holmes et al. [189] showcased further inference attacks via light emanations from a screen-detected from a smart device such as a smartwatch can be used to determine the distance of the wrist from the screen and therefore the keys being pressed. This can also be used in conjunction with database information to detect and identify potential webpages from the light emanations. Leading to a violation of user privacy and circumnavigating existing security protection policies.

Furthermore, Maiti et al. [190] demonstrated the attack surface of Smart Bulbs, violating the target’s privacy and security through inference attacks on audio and video playback from the multimedia-virtualization functionality of the Smart Bulb. They also exposed the possibility of covert channels out of a secured network using the bulbs' visible and infrared light characteristics for data exfiltration. Consumer Smart bulb technology is not considered a common attack vector, meaning that users are less likely to take security precautions as they potentially would with other scenarios such as password protection. Earlier work related to smart bulb’s functionalities by Ronen et al., [191] highlighted that on top of data exfiltration, researchers were able to create strobes of light at epilepsy-inducing frequencies. This means that as well as PII inference leakage, physical harm could potentially be done to users by malicious actor

Just like accelerometers and gyroscope motion sensors can be used to infer a user’s location, the same can be done with light as a metric. Demonstrating this, Kuo et al., [192] used a commercial LED light’s ability to perform optical pulses that are imperceptible to humans, to create an indoor localisation system. They showed that in conjunction with a smartphone’s ALS, accurate indoor localisation can be achieved via multiple light sources. Other works have replicated this system [193]. Work by Xie et al., [194] achieved similar results. Their ‘Light Intensity-based Positioning System’ (LIPS) took ALS from smartphones and created an indoor positioning system that could operate with a sole light source, improving on previous work.

Li et al. [195] created ‘Epsilon’ a system that handles multiple LED anchors to communicate across the same visual band to allow for localisation, obtaining accuracies of 0.4m, creating a system viable for localisation in large indoor spaces such as warehouses. Rajagopal et al., [196] use a similar system using a smartphone camera’s shutter speed to communicate between the device and modified LEDs to provide localisation. Multiple works use various methods to access modified LEDs [197], [198], [199], [200]. With some other works tackling innate issues such as tracking with uneven distribution of light sources [201]. Zhang et al., [202] created PIXEL, a system that archives indoor visible light positioning (VLP) localisation by addressing uneven sampling rates with adaptive downsampling and computational optimization algorithms caused by low-cost equipment.

While the majority of light-based localisation systems that can operate off a user’s ALS system rely on the sender component to be modified in some way, other works have been able to create indoor localisation techniques using unmodified light sources. Zhao et al., [203] used Navi Light which utilises inertia measurement units (IMU) created by inertia light sensors, being other sensors such as accelerometers and gyroscopes used with ambient light sensors to create a light intercity field (LIF) map. Whereby the IMU can be cross-referenced with the LIF map to show the device location in a premeasured environment. Xu et al., [204] achieve a similar end result using IMU based dead reckoning with location fixes.

Qu et al., [205] in conjunction with the IEEE also present comparisons to the various methods of VLC-based localisation. Further work by the IEEE [206] discusses known issues with VLP technology with the guise of overall standardisation. One of the main issues with VLC and VLP comes from the potential lack of dimensionality in ambient light caused by the simplicity of ALS. Previous works have addressed this through various means, such as using more than one sensor, as in the case of IMU. Wang et al., [207] provide another solution through the use of light spectral information. Through the capture of RBG values, which most ALS have the inherent ability to capture, they were able to fingerprint indoor locations at a higher ability compared to that of one-dimensional light intensity information.

The aforementioned works in this section have shown that while in general light as a medium for the transference of sensitive information has been well documented, the sole use of ALS present on smartphones for location inference through the use of digital ‘Fingerprinting’ environments is lacklustre. In most cases, this is easily done via the implementation of multiple sensors, such as was done in work by Azizyan et al., [170]. Fundamentally, as shown by the aforementioned works, the main factor hindering security comes from the fact that on Android the ALS system can be utilised without express permission from the user [208], leaving the device vulnerable.

Chapter 3: Methodology

In this section, the methodology of the project is outlined. This chapter provides a detailed explanation of this project's research design choices, its data collection method rationale and its data processing techniques. The primary goal of the project is to create and establish a systematic and methodological way in which ambient light sensors on smartphones can be used to infer information that can lead to device localisation through digital fingerprints of various locations. This will answer the research question "*How can smartphone ambient light sensors (ALS) be exploited to infer environmental information and what measures can be taken to mitigate the potential privacy and security risks*"

3.1 Research Design

This paper uses a mainly quantitative approach to answering the research question, combining the numerical analysis of light data, signal processing, and preliminary classification through the use of machine learning with qualitative insights into the potential privacy and security risks. Qualitative methods are also used to create a mixed-method approach.

3.2 Project Architecture Design

This project follows a dual-architecture approach. As seen in Section 4.1. The first being the ALS recorder application and the second element is the data processing platform. This architecture was designed in this way for several reasons, but primarily because of the impracticalities of presenting and performing data processing on a smartphone device. This modular approach also allows for a more streamlined approach to making changes or improvements during the development process. In addition, as machine learning techniques were implemented processing on a larger workstation becomes more practical.

3.3 Data collection

3.3.1 Theoretical Rationale

The rationale behind this project is that as smartphones have progressed technologically, so too have their use and their prevalence in everyday life. This means that smartphones have increased interaction with information of all types including sensitive information. They also have many sensors which allow them to interact with their environment more efficiently, but this can also mean that they "know more" than the user recognises. Onboard sensors include amongst many others (see section 2.5.1) an ambient light sensor (ALS). This is more commonly used to control screen brightness under various lighting conditions to save battery power. However, as shown in the literature review (Section 2.6) there is a lack of literature surrounding the sole ALS and the possibility of information leakage. This project is focused on the fact that the security and privacy implications for localisation have not been adequately studied.

3.3.2 Selection of Tools:

The device used in this project is the Samsung Galaxy S10 plus (SM-G975F/DS) [209] running Android 12 also known as Snowcone. This device was chosen because Android 12 is one of the most common Android operating systems in use [209] and despite its age, the S10 Plus is still quite popular [210]. This then reflects a real-world situation. In addition, the availability of a device for experimentation was taken into account.

3.3.2.1 Smartphone Selection

The Samsung Galaxy S10 Plus has its ambient light sensor integrated within the Android sensor stack. As shown in Figure 8, the Android open-source project (AOSP) stack is the standardised framework that allows and controls the interactions between the devices' software components and hardware components. Sensors are the specific hardware this paper is concerned about.

The Android stack includes security protocols and measures that govern how hardware such as sensors are interacted with by applications. For example, the physical ALS sensor the "AMS TCS3407 Uncalibrated Lux Sensor" [211] as used on the device is capable of creating RGB values however due to the security limitations of the stack only the lux value is exposed.

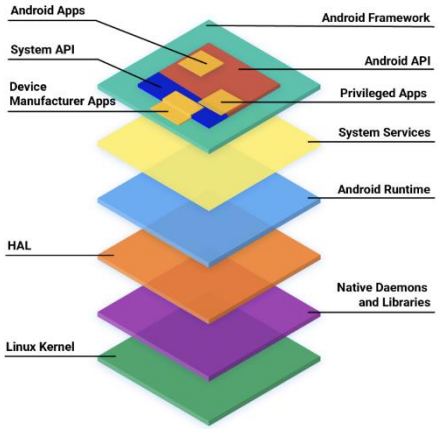

Figure 8: AOSP android architecture stack - [213]

3.3.2.2 ALS Recorder Application

The mechanism for recording the ambient light is a custom-created Java-based application. This application was designed so that the raw lux data could be formatted in a JavaScript Object Notation (JSON) format. While there also exists ALS recording applications on the Android Play Store [213], using a custom-built application provides more control of the details of the application. For example, the ability to create custom metadata was crucial.

3.3.2.3 Data Storage Format

The rationale for the output of the ALS recorder application being in the JSON format is that JSON is understood as an industry-standard communication format between applications. This is mainly because JSON is text-based but is human-readable and easily understood by computers.

3.3.2.4 Inter-System Communication

A way is needed for the two systems to communicate and there are many different approaches for inter-system communication. For example, these approaches include manual file transfer, cloud-based transfer, and automated file transfers such as the Android Debug Bridge (ADB) [214]. This project made use of ADB, which is a command-line tool that implements a background process (daemon) on each system to facilitate communication. This system was chosen due to its versatility and efficiency.

3.3.3 Location Selection and Data Acquisition

3.3.3.1 Location Selection

As the basis of this project is using the ALS for differentiating between locations, the location selection criteria were important. In total, there were 5 locations chosen to give ample opportunity for enough variability in datasets to allow for different patterns to be detected. This project focuses on indoor localisation with 4 out of 5 locations are indoors. The final location, situated outside was chosen to show the disparity in indoor and outdoor locations. Each location was chosen based on its individual physical and light attributes. For example, each indoor location experiences a stable and constant natural light source.

3.3.3.2 Data Acquisition Protocol

The decision was made to place the ALS recorder in the same stable, stationary location in each room, in order to reduce noise in the data and increase the reliability and consistency of the data. It could be conceivable that in a real-world example, a device would not be stationary, however, having a stationary recording point reduces variability in the data. It is also of note that the recording period was chosen to be 7 minutes. The reason for this is semi-arbitrary, in that having too small a recording period would harm the usefulness of the data and having too long a period would become impractical.

3.4 Data Processing

3.4.1 Rationale

Feature extraction and data visualisation are crucial elements in analysing and identifying trends and patterns in the data for localisation.

3.4.2 Techniques

There are a number of different techniques employed in this project. These being, data visualisation, key feature extraction, and also machine learning, which will be outlined in this section below.

3.4.2.1 Data visualisation

For data visualisation, Time-series graphs were chosen to represent the light-intensity data. This provides a visual representation, allowing for a basic understanding of the data to take place. As an example, it can be seen if a location is either inside or outside due to the increased lux values. With inside being less than 500 lux and outside locations during the day being in excess of 1000 lux. Another positive of this technique is that anomalies or interference from natural phenomena can be easily seen. Such as a cloud or other temporary obstruction.

3.4.2.2 Key feature extraction

As each data set might seem different when compared to others taken in the same location, they share similar attributes or features. These are usually governed by the environment they were taken in. For example, data taken from a room with one natural light source compared to a room with a different layout and artificial lights will have a different underlying common factor the datasets will share. The amount of information available is limited to the lux value. As well as the lux value's relationship to itself and the timestamp. However, disambiguation is still possible, for example, one data set would have different minimum, maximum and mean range values compared to another. In an effort to achieve a higher level of distinctiveness, this project uses many key features as outlined in section 5.1.

3.4.2.3 Machine Learning

The goal of this project is to analyse the data taken from multiple different environments and be able to develop and ascertain the intrinsic digital signature that is unique to its environment. This digital signature consists of the key features within a given dataset, representing the unique characteristics of each location. With a developed signature that represents a location, specific environments can be identified from new data, even when there as small variations in the data caused by differing times of day or the device being in a slightly different location in the environment. A possible avenue for gathering and comparing an environment's digital signature is through the use of machine learning. For example, machine learning is used in other industries, such as in biometric identification by analysing a face's biometric digital topography and creating a digital signature for comparison [215].

Supervised classification machine learning algorithms [216] are a sub-class of machine learning algorithms specifically designed to take labelled data and train against it to create models to predict the class of new unlabelled data. This approach is well suited to this project since the task is to assign incoming ALS data to a specific environmental category. In section 3.4.3.3.3 the machine learning classification models are further outlined.

3.4.2.3.1 Classification Model Selection

The choice of which classification models to implement depends on a number of factors [217], the type of data, the quantity of data, the data complexity and the end goal play a factor in which model to apply to a given data set. As there are many different classification models, each with its own positives and negatives, experimentation is important. In this work, various classification models were experimented with, see section 5.2. However, these aforementioned considerations apply to the datasets as follows:

- **Data Type:** Since the data consists of continuous light intensity values with a possible high degree of variability, models [217] such as Naïve Bayes and Logistical Regression are well suited.

- **Data Quantity:** The size of the data set matters, for example, some classification models, like Naive Bayes and J48 Decision trees require less training data. However, with larger data sets, models such as SMO and Random Forest are able to capture more nuanced patterns [217].

- **Data Complexity:** Data complexity also must be accounted for. For example, the overall range of lux values plays a factor as well as the sub-category lux value range. Meaning some models would find it easier to account for the disparity in lux values between indoor and outdoor environments. Multilayer Perception models are a good candidate for this as they are able to handle complexity through recognising non-linear patterns in data [218].

- **End Goal:** as the end goal for this project is to accurately classify new data into specific environments, models that are able to handle unseen data are advantageous. Logistic regression and Random Forest models are good candidates due to their ability to handle variable data conditions [217].

3.4.3 Selection of Tools

For data processing, there exist a number of different tools to help achieve the aforementioned techniques. Therefore, the rationale and methodology behind these decisions are crucial as each technical part of the project requires different tools as the individual requirements are different.

3.4.3.1 Programming Language

For both the data visualisation and key feature extraction processes the appropriate software tools must be chosen. While there are many different programming languages, each one has its own use case, and, some languages excel in data processing and analysis [219]. While most commonly used programming languages are both powerful and versatile, the programming language Python is a popular choice [220]. Python allows for the use of many different libraries, such as pandas [221], NumPy [222] and SciPy [223], facilitating data analysis and visualisation.

Another consideration with the choice of programming language is its applicability in machine learning. Python has access to machine learning-centric libraries such as Scikit [224] and while not a common library, Weka [225], which is used for calling Weka classifiers. Python also is a popular open-source language, meaning it has a large community and many learning resources surrounding it. Hence this project makes use of Python.

3.4.3.2 Software Platforms and IDEs

The choice of the different software platforms is closely related to the rationale for the choice of programming language. For example, as this project is using Python, there are a number of platforms and integrated development environments (IDEs) that could be used, such as Visual Studio [226], PyCharm [227] and Google Colaboratory (Colab) [228]. This project uses Google Colab, which is a software platform that implements an adaption of Jupyter Notebook [229], allowing users to write and execute Python code via a web browser. This platform was chosen for a number of reasons, the main reason being that it is a cloud-based platform. This means that the computation is performed in the cloud rather than client-based allowing for external computing resources to be used. Which is useful for performing intensive machine-learning computations.

3.4.3.3 Weka

There are many different software packages tailored to carry out machine learning computation, For example, KNIME [230], Azure [231], and Weka, which are all commonly used for machine learning, data analysis and exploration. This project makes use of the Waikato Environment for Knowledge Analysis (WEKA) Platform for machine learning exploratory analysis. The rationale for choosing Weka was because of its comprehensive data preprocessing capabilities, where built-in tools can perform tasks like data cleaning automatically. Weka also has a built-in third-party Python wrapper [232] allowing for Weka to be called in other environments, such as the aforementioned Google Colab.

Weka is separated into 5 distinct internal applications, the Explorer, Experimenter, KnowledgeFlow, Workbench, and Simple CLI, which fall into three categories: data set processing, machine learning schemes, and output processing [233]. While each internal application is distinct in many cases their use cases overlap. For example, data set processing can be performed in the Explorer as well as the KnowledgeFlow application.

Data set processing allows the user to extract information from a given dataset, as well as perform data processing operations such as splitting datasets into training and test sets. Data features can also be removed through filters, thus enabling a dataset to become suitable for machine learning.

The machine learning category of internal programs allows the user to implement a list of various machine learning algorithms. The output processing category takes the output from a given machine learning schema and performs tasks with it. For example, evaluating rules against a test model.

3.4.3.3.1 Weka data format

The attribute-relation file format (.arff) was developed by the Machine learning project at the University of Waikato to be used with the Weka software platform [234]. This file format is considered a simple and easily readable format that is used to describe a list of instances which share a set of attributes. The file format consists of two sections, the header and the body. The header contains information such as the data types and the names of the relations and the body contains the actual instances of the data as rows.

This project discusses the implementation of .arff files and the conversion between JSON and .arff file formats in Section 4.3.3.1. While Weka does support other file types such as CSV [235], Weka is purpose-built around the .arff file format so conversion to this format would be advantageous for the project for compatibility reasons.

3.4.3.3.2 Weka machine learning schemes

Machine learning schemes can be considered implementations of a given machine learning algorithm. Weka makes use of various types of machine learning schemes. For example, supervised, unsupervised and meta-learning algorithms are made available to the user. Under these classifications, there are many subcategories of machine learning algorithms available. Table 5 shows what machine learning schemes are available and their descriptions.

Table 5: Weka Machine Learning Schemes

Sub-categorisation	Description
Bayes	Bayes contains learning algorithms that implement the Bayes' Theorem [236] at a fundamental level. Such as Naive Bayes.
Functions	Functions contain learning algorithms that attempt to find a function that can be used to map or estimate the dataset. Such as Linear Regression
Lazy	Lazy contains learning algorithms that make use of lazy learning to perform classification. Such as Instance-Based k-nearest neighbour (IBk), which is an implementation of the k-nearest neighbour classification algorithm
Meta	Meta contains algorithms that consist of a combination of different machine-learning models. Such as Bootstrap aggregation (Bagging)
Misc	Misc contains algorithms that do not fit in other categories or are unique in their approach. Such as wrappers that deal with incompatibilities between training and test data sets
Rules	Rules contains algorithms that adhere to a set of rules to make a prediction. Such as Decision Tables, which are structured sets of multiple conditions
Trees	Trees contains learning algorithms that are based on decision trees. For example, Random Forest makes use of n-levels of decision trees to make predictions.

3.4.3.3.3 Evaluation Methods

Weka uses many different ways to evaluate the success or applicability of a classification algorithm [237]. While all the available metrics offer insights into the data, certain metrics are more relevant for this type of dataset. The accuracy rating metric is one of the more fundamental that measures the proportion of correctly classified instances. The Kappa statistic is also fundamental in that it measures the agreement between predictions and the actual classifications. The precision and recall metrics are also very important to help understand and evaluate the chosen model's ability to correctly identify positive instances against false positives and the F-Measure combines the precision and recall to give a total model evaluation score.

3.5 Ethical Considerations and Limitations

In this section, the ethical considerations and limitations of this project are addressed. In research, the ethical considerations taken are crucial to ensure that the overall study respects the key ideas of CIA as well as privacy.

3.5.1 Ethical Considerations

This project was conducted without any human subjects, thereby ensuring that no sensitive or personal information could be disseminated or later collected and analysed. While the five environments in which the experiments take place are real physical locations, no identifiable information has been exposed. The focus of the text is solely on the interactions of ambient light with the ALS, and while the produced digital signatures do contain unique identifiable information it cannot be used to violate privacy.

The information contained and recorded on the ALS recording device was secured to prevent unauthorised access, through the use of access control (ISO/IEC 27001 Section A.9) [238]. However, since the risk and impact are minimal because of the usefulness of the captured ALS data excessive steps to secure the data are unnecessary.

3.5.2 Limitations

This project faced several limitations in general. Firstly, this study was conducted between the months of April, May, and June. This is only a short time span as recording over multiple months in different seasons would help identify and reduce the influence of the relative seasonal light on the ambient light of the environments. For example, natural light patterns change throughout the year, and the average angle of incidence on the ASL sensor fluctuates. Also of note is that the recording periods spanned 7-minute periods once a day for each location, meaning that whole scenarios were not captured such as dawn, dusk, and nighttime. Also, short 7-minute periods of recording could be susceptible to environmental phenomena influencing the results, such as clouds blocking out the sun.

As mentioned in section 2.6, device hardware irregularities could unintendedly influence the results. Only one device was used, therefore the impact of this is unknown. Another limitation is that the environmental conditions were strictly controlled, which would not reflect a real-world scenario. For example, no artificial lights were used, only natural ambient light. Another limitation is that of the dimensionality of the data, having only the lux value and its relation to its timestamp for information does not provide as much analysable information as RGB values as an example. Finally, while it is true that the recording device's "AMS TCS3407 Uncalibrated Lux Sensor" [211] is RGB enabled, these values were inaccessible and unexposed through the Android Sensor Manager API [239] due to its security policies. However, these values could be accessed but this process would involve the circumnavigation or disabling of Android security systems such as SELinux.

Chapter 4: Implementation

In this section, The ALS recorder design and implementation will be outlined. This was used to measure the ambient light in various locations to uncover an environment's digital signature

4.1 Experiment Outline and Setup

Figure 9 shows the layout of the experiment. The experiment employs the use of the smartphone (Samsung Galaxy S10+) as described in section 3.3.2.1, running the ALS recorder application, (Section 4.2) which stores the data in the application's external storage using the Android file system architecture. The smartphone is placed in the different environments outlined in Table 6, where the ambient light levels are recorded and stored as JSON files. After using Android Debug Bridge to access the stored files. An external device is used to process and perform data visualisation, in which later analysis is performed as shown in section 5.1.

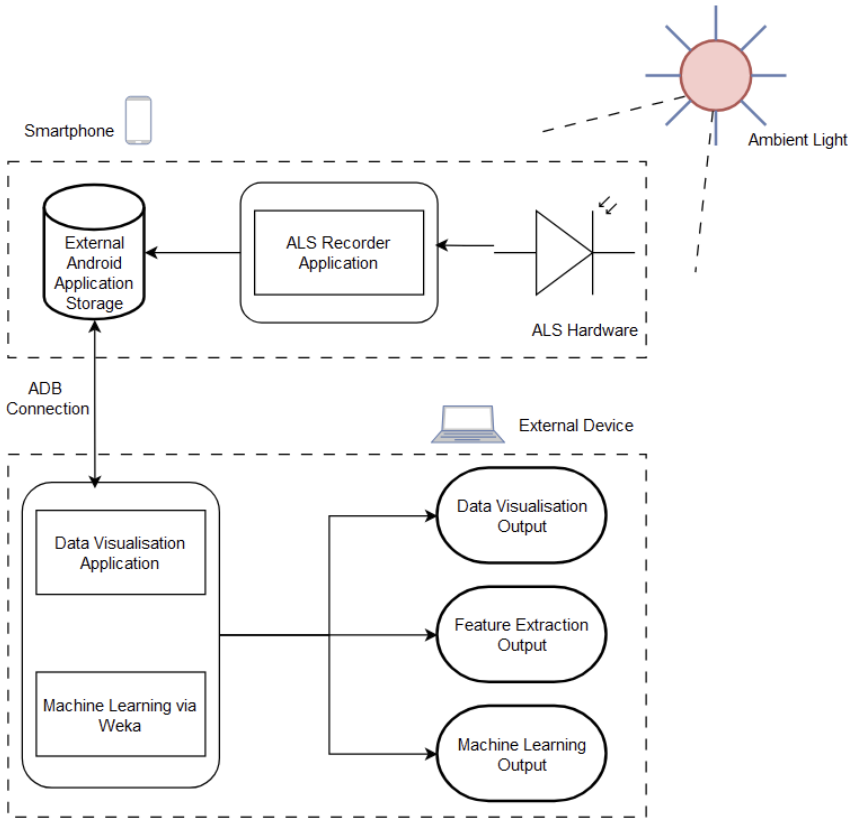

Figure 9: Experiment setup

In order to capture the ambient light fingerprints of different locations the Smartphone is placed in multiple different locations as shown in Table 6.

Table 6: Tested Locations

Environment Location	Environment Description
Landing	An intermediary hallway area between rooms
Living Room	A room with a mix of natural light and artificial light.
Kitchen	A kitchen area with a wider range of natural light access
Bedroom	A standard bedroom with natural light access.
Outside	An outdoor location with no obstructions to natural light

4.2 ALS Recorder Application

The ALS Recorder application is a Java programme that allows for access to the smartphone's ALS device and allows the user to input 'locational data' to serve as metadata providing context to the JSON file. Figure 54 shows the UI of the ALS Recorder application.

4.2.1 Application Design

Figure 54 shows the 'Ambient Light Sensor Recorder', which is in a developmental stage, therefore the UI has been built for the core functionality of the application. Which is to record ambient lux data and convert it to a JSON format while allowing the user to append metadata in the form of 'Locational Information' which is used to provide environmental information and context. Meaning that effective data collection for analysis is possible.

4.2.1.1 UI Features

Figure 54 shows the Ambient Light Sensor Recorder's user interface features. These features include the real-time lux data display, allowing the user to monitor the lux level in real-time. The next feature is the Start and Stop recording buttons, these simply just begin or end the recording period depending on the need. Finally, the 'Locational Information' field allows for the user to input any metadata information such as the location of the recording as well as the time, date and any notable conditions, such as weather.

4.2.1.2 UML Diagram

The UML Diagram presented in Figure 10 provides a visual overview of the recording application's internal structure and data flow. Below the relationships between the different elements shown in the figure are outlined:

'MainActivity': This is the core component of the application, which handles the user interaction and the recording process. Which is initiated by the user. It also handles the user inputs from the metadata field. As this is the core component, the other classes are called from here too.

'SensorManagerHelper': This class is instituted by Android as a way to handle interactions with the device's ALS hardware. For example, it handles the deregistration and registration of sensors and manages the settings controlling aspects such as the sampling rate used for recording.

'SensorDataHandler': This class manages the storage of the captured ALS data, ensuring that it is correctly formatted into JSON and stored in the application's shared external storage so it can be easily extracted. This class also handles checking to ensure storage availability and error checking.

'Android.Hardware': the 'Android.Hardware.SensorEventListner', 'Android.Hardware.Sensor', 'Android.Hardware.SensorManager' are all Android framework functions and classes that provide the interface for interacting between the application layer and the device's hardware through the device drivers

Figure 10 is a UML diagram showing the internal data flow of the ASL recording application.

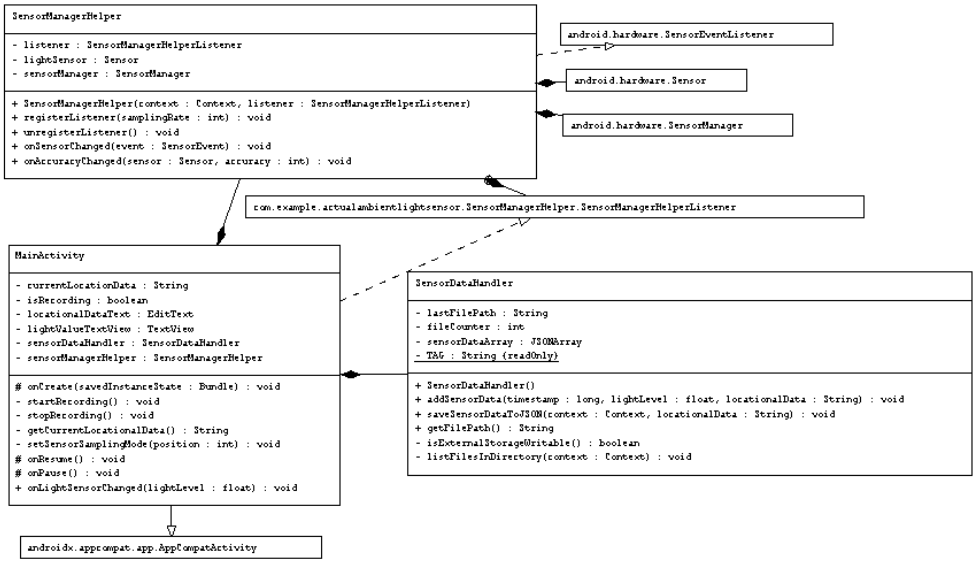

Figure 10: ALS Recorder Application UML Diagram

4.2.2 Component Diagram

Figure 11 shows the component diagram, which details the internal interactions of the ALS recorder application. The diagram presents the components outlined in section 4.2.1.2 as well as other system-level components such as ‘ExternalStorageManager’. The main ALS Recorder Application interacts with the ‘SensorManagerHelper’ and ‘SensorDataHandler’, which are in control of managing data storage and sensor interactions via the Android hardware sensor manager and the ‘ExternalStorgaeManager’ respectively. Thus, showing the data flow of the application.

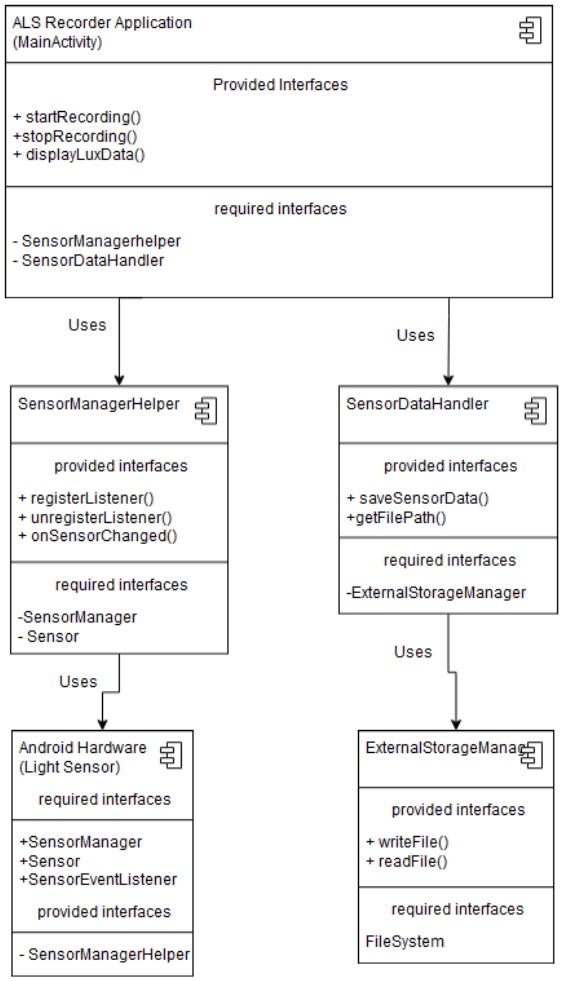

Figure 11: Component Diagram

4.2.3 Data Storage

Figure 12 shows an example of the JSON format created by the ‘Ambient Light Sensor Recorder’ The JSON key-value pairs used in this project are as follows: ‘timestamp’ in the UNIX time format, ‘lightLevel’ in lux, and ‘Locational Data’ being the custom metadata following the day, hour, minute, location format.


```
[{"timestamp":1716984783224,"lightLevel":74,"Locational Data":"29:13.12_Bedroom"},
{"timestamp":1716984784124,"lightLevel":75,"Locational Data":"29:13.12_Bedroom"},
{"timestamp":1716984784424,"lightLevel":76,"Locational Data":"29:13.12_Bedroom"},
{"timestamp":1716984784724,"lightLevel":78,"Locational Data":"29:13.12 Bedroom"}]
```


Figure 12: JSON Data Format

4.3 Data Processing

The section below outlines the data processing. This consists of three elements, data visualisation, data feature extraction and then using machine learning classification techniques on said data features.

4.3.1 Data Visualisation

Figure 13 shows the logic flow for creating a visual representation of each location.

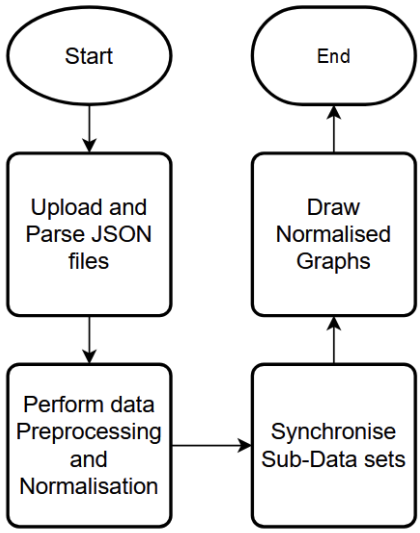

Figure 13: Data Visualisation flowchart

4.3.2 Data Feature Extraction

Figure 14 shows the logic flow for processing the ALS data and extracting a presenting signal feature, these features are presented in section 5.1

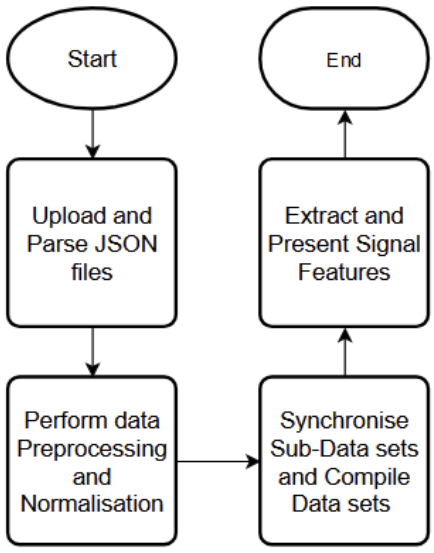

Figure 14: Data Feature Extraction flowchart

4.3.3 Machine learning Implementation

As described in section 3.4.2.3, machine learning was implemented in this project to perform environmental classification on the captured ALS dataset.

With the methodology for this project's machine learning being outlined in section 3.4.2.3, this project implements various machine learning algorithms to perform environmental classification with varying degrees of success through the use of Weka. As shown in section 3.4.3.3, Weka provides multiple types of classification algorithms. Table 12 presents 16 different classification algorithms that were applied to the ALS environmental dataset using the default Weka settings. out of the 16 classification models tested, the 3 best performing are presented in section 5.2. Weka also provides pre-processing methods, such as cross-validation and automatic test and training data set splitting.

The best-performing classification algorithms were, Naïve Bayes [240], J48 [241], and Logistic Regression [242]. These classification algorithms in general demonstrated a higher level of accuracy compared to some of the others. However, due to the nature of the dataset, as shown in Figure 32, the outside dataset was correctly chosen with 100 per cent accuracy. This means that accuracy scores presented with the outside data do not truly represent the viability of the models.

As a way to present this fact the same classification models were run with the omitted outside dataset as shown in the second column of Table 12. Also, of note is that as this was a preliminary approach to machine learning, models could be improved through various means, such as better tuning, through the use of hyperparameter optimisation [243], further explored in section 6.2. Depending on the needs of each classification algorithm and how they interact with the data. Also, the combination of multiple classification algorithms through ensemble learning could potentially lead to better results.

4.3.3.1 File Conversion

As described in section 3.4.3.3.1, Weka utilises the purpose-built .arff file format, and since the data collection application outputs a JSON file, conversion is needed. Figure 15 shows an example of the .arff file format used. Manual editing was performed through the find and replace function on Notepad++ [244] to achieve file conversion for compatibility.

```
@relation testdata

@attribute timestamp numeric
@attribute lightLevel numeric
@attribute 'Locational Data' {Kitchen,Bedroom,Landing,Outside,LivingRoom}

@data
1716296833242,26150,Outside
1716296833413,26137,Outside
```

Figure 15: .arff file implementation

Chapter 5: RESULTS

This Section will outline the results of the project, which entails the digital representation and digital features of each room. Then various classification models are outlined and evaluated for efficacy.

5.1 Data Visualisations and Analysis

The following sections are separated by the testing environments and entail the raw data visualisation's location images and outlines of said locations. Each recording of ambient light consists of 7-minute-long sessions from a stationary position as indicated. These raw data visualisations help to gain insight into the data and understand the data variations, patterns and anomalies. We will also explore the extracted digital key signal features from each localised environment to help with localisation differentiation.

5.1.1 Bedroom

Table 7 shows Figures 16 and 17, representing the bedroom environment. Situated away from the possibility of direct sunlight and away from direct sources of natural light. The device was also placed on a bedside table, a situation which reflects possible real-world conditions. The ALS recording test was performed without the use of any artificial light sources.

Table 7: Bedroom Environment

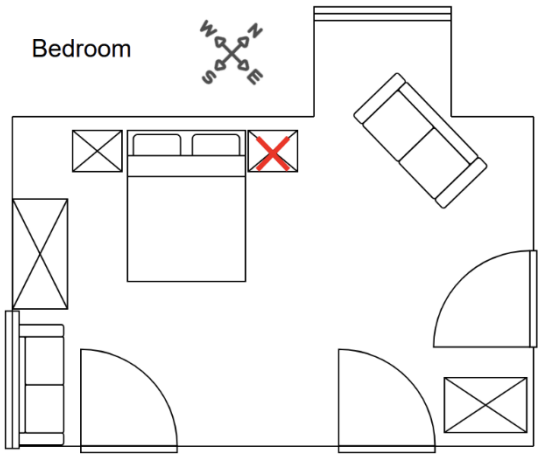

Figure 16: Bedroom Layout

Figure 17: Bedroom

Figure 18 represents the raw ALS data recordings from the bedroom environment. The lux data graph demonstrates some of the issues with the experiment. ‘Data 8’ as shown below is an outlier. This could be due to many reasons, as explained in section 3.5.2.

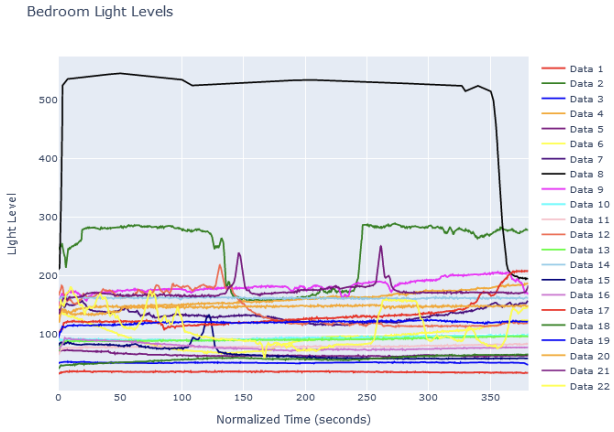

Figure 18: Bedroom Measured light levels

5.1.1.1 Bedroom digital representation

Figure 19 below represents the extracted key signal features from the ALS data across multiple datasets for the Bedroom environment. The 22 datasets that constitute the bedroom digital representation are characterised by features such as the number of data points, the average light levels, the variability in light levels shown in the data variance and standard deviation. The table also represents the range of each data set and the skewness and kurtosis showing the distributions asymmetry and its tail heaviness. These digital features are essential in allowing for the differentiation of the different environments.

Extracted Signal Processing Features:

dataset_index	num_data_points	mean_light_level	std_light_level	min_light_level	max_light_level	median_light_level	light_level_range	light_level_variance	light_level_skewness	light_level_kurtosis
0	2833	35.581186	0.982531	29.0	38.0	35.700000	9.0	0.965367	-0.964859	2.220636
1	3025	247.852562	49.193399	158.0	289.0	277.000000	131.0	2419.990530	-1.021168	-0.812751
2	2740	51.327372	0.673779	47.0	53.0	51.384615	6.0	0.453979	-0.410482	1.755861
3	3048	164.379429	17.114216	132.0	207.0	161.416667	75.0	292.896385	0.858366	0.125927
4	2817	64.192758	2.753269	61.0	74.0	63.307692	13.0	7.580489	1.823756	2.345358
5	2771	95.336702	17.921908	67.0	137.0	96.000000	70.0	321.194764	0.116429	-0.856815
6	3025	133.663306	11.060808	115.0	157.0	132.266667	42.0	122.341474	0.201319	-0.932601
7	3060	476.058497	121.558688	193.0	545.0	528.815538	352.0	14776.514732	-1.792239	1.291466
8	3001	181.506498	11.337513	143.0	207.0	179.000000	64.0	128.539197	0.410681	-0.423913
9	2932	93.500682	3.070353	88.0	100.0	93.333333	12.0	9.427068	0.204322	-1.395655
10	3035	79.573641	2.351585	65.0	86.0	78.777778	21.0	5.529953	0.412023	2.986579
11	2985	138.220771	27.172142	112.0	218.0	121.000000	106.0	738.325301	0.694612	-1.088105
12	2774	91.109409	2.872327	72.0	97.0	90.500000	25.0	8.250260	-0.282102	3.369205
13	3033	161.555226	0.946151	156.0	165.0	161.500000	9.0	0.895202	-0.420152	1.784677
14	3008	67.221576	12.766604	52.0	132.0	59.950428	80.0	162.986176	1.950010	5.202114
15	2785	78.164452	5.586147	68.0	94.0	76.111111	26.0	31.205043	1.121495	-0.029003
16	2915	136.596055	27.732006	103.0	212.0	126.166667	109.0	769.064152	1.735189	1.705601
17	3072	58.600260	4.943118	41.0	67.0	58.500000	26.0	24.434417	-0.520047	-0.312868
18	3104	120.370168	3.077927	93.0	126.0	120.857143	33.0	9.473633	-2.032707	10.733272
19	2759	145.712939	4.320840	120.0	152.0	147.000000	32.0	18.669660	-2.199681	4.629404
20	3033	174.257501	13.353836	134.0	250.0	170.666667	116.0	178.324941	2.719752	9.749471
21	2828	110.909653	30.654387	60.0	181.0	96.763889	121.0	939.691429	0.505380	-1.190143

Figure 19: Bedroom Signal Features

5.1.2 Kitchen

Table 8 shows Figures 20 and 21, representing the kitchen environment. The device placement is situated away from direct sunlight. The majority of the room's different sources of natural light are west-facing, meaning that during the time of the experiment, there was no direct sunlight giving a more accurate representation of the ambient light. The ALS recording test was performed without the use of artificial light sources.

Table 8: Kitchen Environment

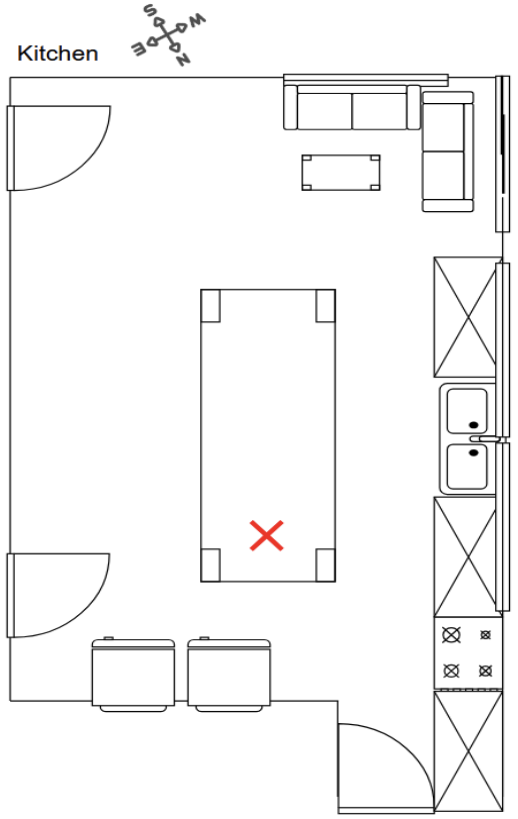

Figure 20: Kitchen Layout

Figure 21: Kitchen

Figure 22 represents the raw ALS data recordings from the Kitchen environment. The graph is characterised by days of low volatility and days with high volatility. As this experiment relied solely on the use of natural light sources, then natural phenomena, such as clouds, could cause or exacerbate the volatility within the data this is further explained in section 3.5.2

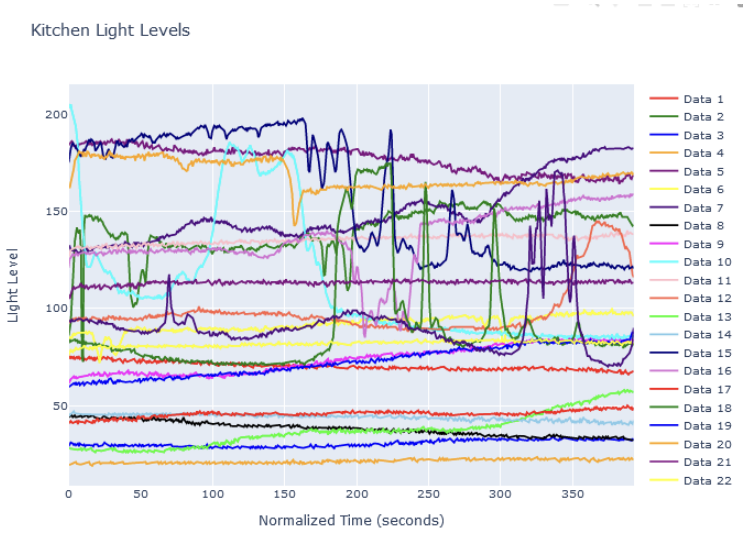

Figure 22: Kitchen Measured Light Levels

5.1.2.1 Kitchen Digital representation

Figure 23 below presents the extracted signal features from the Kitchen ALS dataset. The 23 data sets are each broken down and have their signal features shown. The number of overall data points, the average across all light levels, and the data variability are presented. The table also presents other statistical key features, such as kurtosis and skewness, which represent the asymmetry of the data and its appearance. The differences in these features culminate in the ability to disguise between different environments as the common factor all these statistics share would be the digital 'fingerprint' of the kitchen environment.

Extracted Signal Processing Features:

dataset_index	num_data_points	mean_light_level	std_light_level	min_light_level	max_light_level	median_light_level	light_level_range	light_level_variance	light_level_skewness	light_level_kurtosis
0	595	69.999580	2.268568	65.000000	75.5	69.500000	10.500000	5.146402	0.235428	-0.220580
1	594	139.566077	11.427715	72.500000	155.0	141.000000	82.500000	130.592668	-1.712794	5.499022
2	552	30.472977	1.672499	28.000000	34.0	30.000000	6.000000	2.797254	0.239583	-1.456143
3	561	21.400921	0.974720	19.000000	23.5	21.333333	4.500000	0.950079	0.050986	-1.218947
4	586	176.237770	6.695475	164.000000	187.0	178.750000	23.000000	44.829380	-0.264707	-1.423068
5	563	91.945086	4.381017	72.000000	100.0	92.500000	28.000000	19.193313	-1.061951	1.952267
6	590	151.538559	16.980941	128.500000	183.0	145.000000	54.500000	288.352354	0.726444	-0.882095
7	602	37.689092	3.748965	32.000000	45.0	37.500000	13.000000	14.054739	0.267539	-1.122599
8	559	73.901610	7.122248	61.500000	85.0	74.500000	23.500000	50.726413	0.088441	-1.526026
9	552	112.554650	33.017903	84.000000	205.0	96.875000	121.000000	1090.181948	1.212070	0.122410
10	593	135.493957	2.239114	126.333333	140.0	136.000000	13.666667	5.013633	-0.561450	-0.240879
11	593	101.249438	16.151149	88.000000	146.0	95.333333	58.000000	260.859614	1.757422	1.629871
12	604	39.005381	10.223820	26.000000	60.0	37.000000	34.000000	104.526488	0.706085	-0.578504
13	563	43.933837	1.529186	40.000000	47.0	44.250000	7.000000	2.338411	-0.615919	-0.573644
14	590	155.551836	31.575421	117.000000	198.0	147.000000	81.000000	997.007237	0.048398	-1.856953
15	583	136.569039	16.061035	85.000000	159.5	134.000000	74.500000	257.956856	-0.742518	0.556018
16	601	46.048253	2.065331	40.500000	52.0	46.000000	11.500000	4.265594	0.170604	0.758487
17	599	90.375070	26.808985	71.000000	174.5	82.000000	103.500000	718.721691	2.177618	3.564045
18	591	73.383249	7.920652	59.500000	88.0	73.000000	28.500000	62.736735	-0.025337	-1.436840
19	607	168.913646	6.747192	142.500000	181.0	168.000000	38.500000	45.524600	-0.099523	-0.183420
20	598	113.264214	1.103591	105.000000	115.0	113.500000	10.000000	1.217914	-2.069854	8.700904
21	597	81.826633	1.278350	76.000000	84.0	82.000000	8.000000	1.634179	-0.469785	0.119707
22	588	95.853883	22.894356	70.000000	171.0	89.500000	101.000000	524.151534	2.117992	3.793995

Figure 23: Kitchen Signal Features

5.1.3 Landing

Table 9 shows Figures 24 and 25 representing the outline of the Landing environment. This environment is characterised by only one avenue of natural light. However, the same issue of some days having more variation due to variable natural light conditions still persists. Also, of note that the light source is north-west facing meaning there was no direct sunlight due to the time of the experiment.

Table 9: Landing Environment

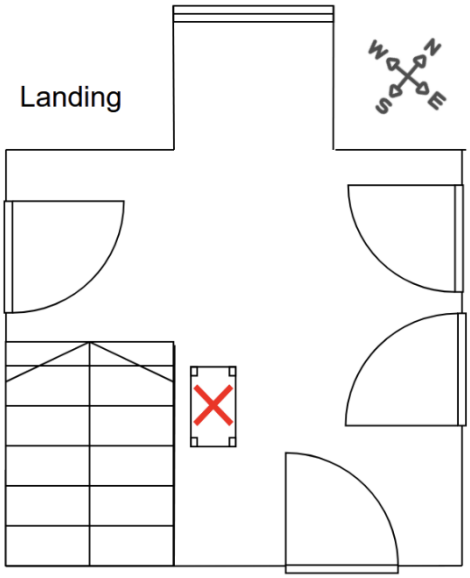

Figure 24: Landing Layout

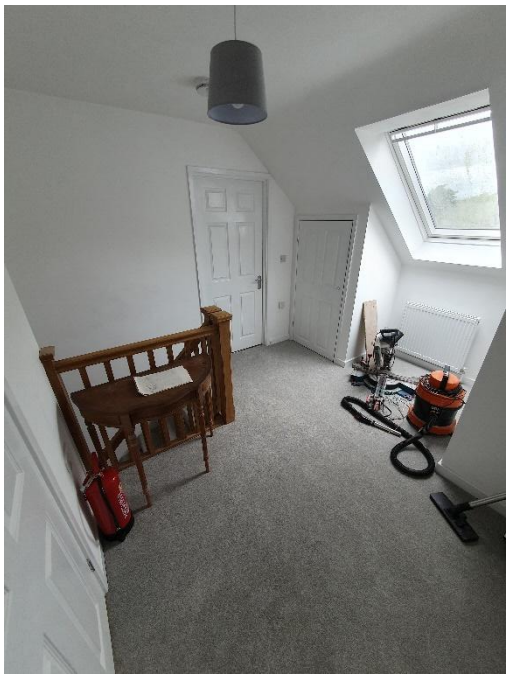

Figure 25: Landing

Figure 26 represents the raw ALS data recordings from the landing environment. This figure has two obvious visual characteristics. The first being that of the more uniform datasets, which are characterised by long periods of low variance. These data sets are most likely to be caused by days in which there was constant thick cloud cover, as the lux values are also comparably low. The second is that some data sets have high volatility, meaning that the natural light levels changed, mainly due to clouds. This is seen by the brief periods of reduced lux measurements, with other possible causes being outlined in Section 3.5.2.

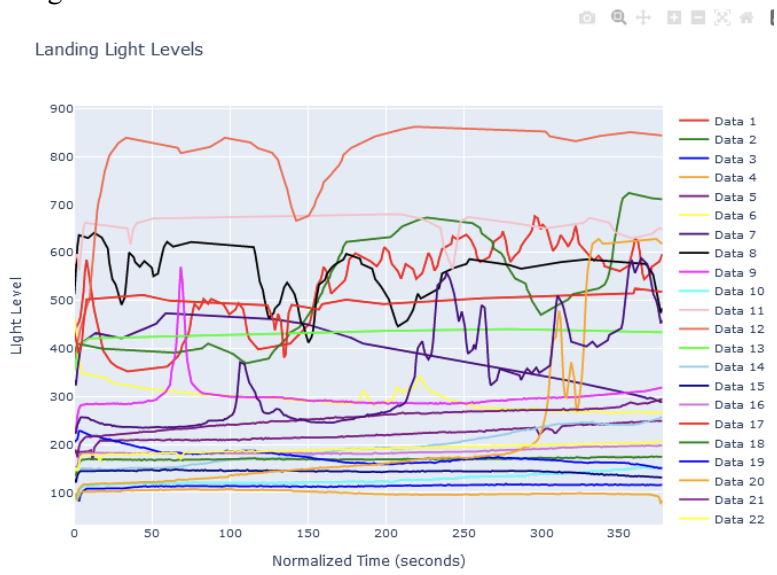

Figure 26: Landing Measured Light Levels

5.1.3.1 Landing Digital representation

Figure 27 shows the extracted key features across the 23 individual datasets from the Landing environment. With each dataset having multiple key signatures extrapolated from it. For example, the figure shows general features such as the number of data points and the maximum and minimum light levels. It also shows more advanced features such as the standard deviation present within the data itself as well as statistical averages and phenomena such as kurtosis and skewness.

Extracted Signal Processing Features:

dataset_index	num_data_points	mean_light_level	std_light_level	min_light_level	max_light_level	median_light_level	light_level_range	light_level_variance	light_level_skewness	light_level_kurtosis
0	430	523.572868	89.456143	352.000000	675.500000	550.750000	323.500000	8002.401546	-0.530901	-0.949808
1	432	540.206019	123.365712	369.000000	724.000000	531.250000	355.000000	15219.098895	0.023744	-1.585147
2	397	112.950042	4.398073	70.000000	118.000000	113.500000	48.000000	19.343046	-5.132406	36.796980
3	388	99.505369	4.465855	76.000000	107.000000	97.500000	31.000000	19.943857	-0.413331	1.353233
4	441	224.638322	17.405556	167.000000	260.333333	221.000000	93.333333	302.953388	0.175098	-0.386320
5	442	290.982466	26.049716	180.500000	464.000000	288.500000	283.500000	678.587699	1.426948	7.088003
6	407	392.817977	61.007138	157.333333	473.000000	400.000000	315.666667	3721.870925	-0.439311	-0.691210
7	439	555.035308	54.248202	329.500000	641.000000	573.333333	311.500000	2942.867425	-0.880853	0.352439
8	425	299.135490	29.571159	202.000000	570.000000	295.500000	368.000000	874.453419	5.622638	42.666618
9	408	128.472426	12.089539	86.500000	155.000000	122.833333	68.500000	146.156961	0.517674	-0.014113
10	426	660.538732	21.178015	512.500000	681.000000	666.964286	168.500000	448.508340	-2.447935	9.249012
11	422	812.474724	76.767549	378.000000	862.000000	835.713942	484.000000	5893.256574	-3.372599	12.713351
12	424	432.239387	8.955211	353.500000	441.000000	433.933173	87.500000	80.195803	-4.610221	33.971234
13	432	203.295910	44.176150	128.000000	342.000000	192.750000	214.000000	1951.532258	0.711774	0.066337
14	434	141.301939	6.090595	104.000000	148.000000	144.000000	44.000000	37.095343	-1.820410	3.597512
15	433	186.348152	7.539352	151.000000	202.000000	182.500000	51.000000	56.841831	-0.060714	2.052966
16	404	499.146452	18.873164	381.000000	525.000000	501.981132	144.000000	356.196309	-3.379753	15.110447
17	426	169.812011	4.282642	125.250000	176.000000	169.500000	50.750000	18.341023	-5.432306	46.943746
18	434	171.294739	18.145561	113.500000	228.000000	168.416667	114.500000	329.261386	0.693267	0.874274
19	432	242.797164	167.940674	91.666667	628.000000	164.000000	536.333333	28204.069933	1.451202	0.575644
20	444	260.219595	25.139377	173.000000	331.000000	264.500000	158.000000	631.988257	-0.150207	-0.055942
21	431	191.773299	9.726270	144.333333	207.000000	192.000000	62.666667	94.600334	-0.769485	1.786927
22	434	342.297523	103.328103	223.666667	588.000000	325.375000	364.333333	10676.696955	0.685867	-0.739262

Figure 27: Landing Signal Features

5.1.4 Living Room

Table 10 shows figures 28 and 29, representing the Living Room environment and the location of its natural light sources The ALS recording smartphone was placed in the centre of the room with no obstructions around it to give it as accurate a reading as possible. The ALS test was also performed without any artificial light sources.

Table 10: Living Room Environment

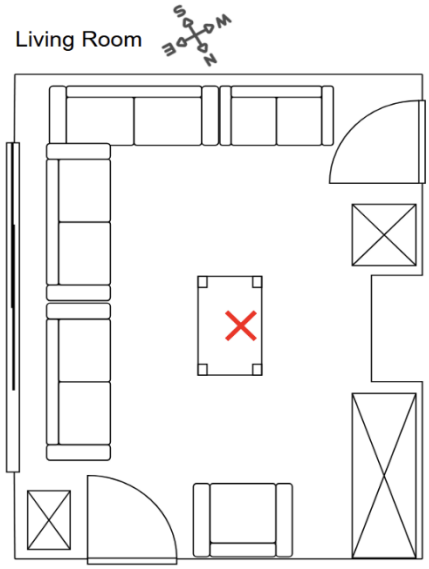

Figure 28: Living Room Layout

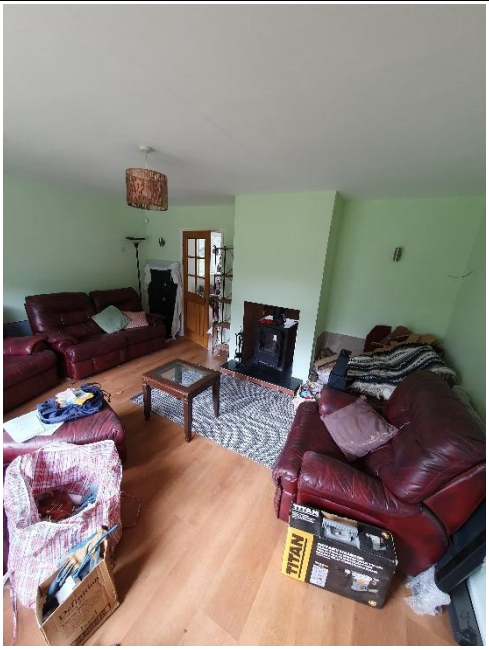

Figure 29: Living Room

Figure 30 represents the raw ALS data recording for the Living Room environment. This figure is characterised by data sets which contain very low volatility and are stable, which indicate little change in the light levels of the environment. And some datasets that contain high levels of volatility. This could be due to sunny days. However, data 1 contains anomalously high lux values. This could be due to any number of reasons, for example, unforeseen reflections causing an increase in ambient light in the environment. Further possible reasons are outlined in section 3.5.2.

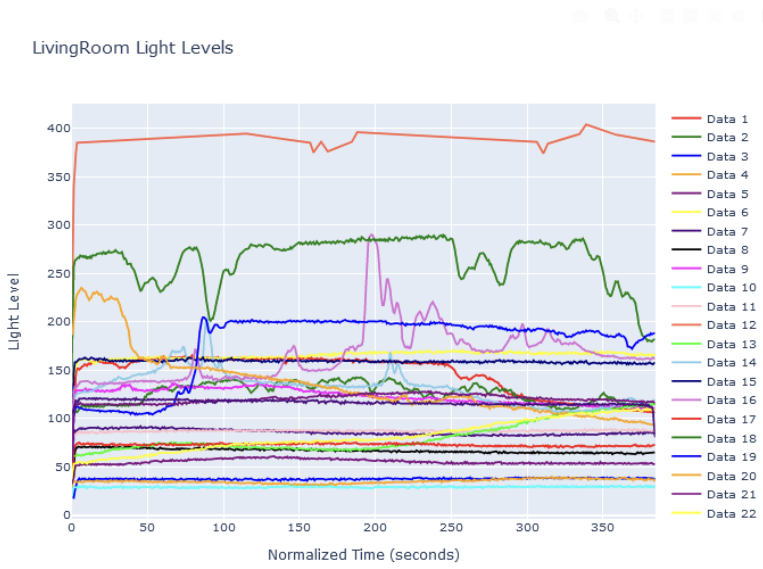

Figure 30: Living room Measured Light Levels

5.1.4.1 Living room Digital representation

Figure 31 below showcases the extracted signal features from the Living room ALS dataset. In the Living Room environment, there are 23 datasets taken, representing 23 different days of recording. There are a number of different key features extracted from each data set. These include the number of data points, the average across the data points, its range with its maximum and minimum values along with each dataset's skewness, kurtosis and variance. These values ultimately provide the unique signature for the room.

Extracted Signal Processing Features:

dataset_index	num_data_points	mean_light_level	std_light_level	min_light_level	max_light_level	median_light_level	light_level_range	light_level_variance	light_level_skewness	light_level_kurtosis
0	585	142.333191	22.927402	83.500000	165.0	157.000000	81.500000	525.665759	-0.691516	-1.226301
1	581	125.574727	10.194692	99.000000	142.5	128.000000	43.500000	103.931745	-0.105775	-1.307066
2	541	37.420826	1.763513	18.000000	39.0	37.500000	21.000000	3.109978	-7.815892	77.597475
3	535	35.086293	2.136880	25.666667	39.5	35.000000	13.833333	4.566256	-0.124332	0.012859
4	585	55.127208	2.829462	32.000000	61.0	54.000000	29.000000	8.005857	-1.422316	11.748811
5	600	165.124028	3.426660	137.000000	170.5	166.000000	33.500000	11.742002	-2.190886	12.545167
6	585	85.700570	3.482339	41.500000	92.0	85.000000	50.500000	12.126686	-4.209639	50.617863
7	569	66.367311	3.112312	39.000000	71.0	66.250000	32.000000	9.686487	-3.932876	31.007168
8	583	122.110206	8.578390	77.500000	137.0	121.500000	59.500000	73.588771	-0.335238	0.026103
9	590	29.109605	0.895133	17.500000	31.0	29.130682	13.500000	0.801264	-6.573216	80.079531
10	587	86.953719	2.448321	44.000000	89.0	87.250000	45.000000	5.994277	-12.641749	193.684555
11	554	388.814681	11.033040	217.666667	404.0	389.840087	186.333333	121.727961	-10.252958	139.234014
12	582	83.779210	18.021161	45.000000	120.0	74.000000	75.000000	324.762239	0.698394	-1.046634
13	586	131.235353	17.041522	65.000000	204.0	132.916667	139.000000	290.413467	0.528347	2.843006
14	547	158.651889	4.335578	96.500000	163.0	159.000000	66.500000	18.797239	-11.022334	140.007505
15	581	166.044607	28.457146	79.000000	290.0	163.500000	211.000000	809.809168	1.605137	4.310008
16	549	72.555859	2.343327	39.000000	75.5	73.000000	36.500000	5.491182	-8.670309	106.549675
17	606	257.103685	31.866632	120.000000	289.5	270.750000	169.500000	1015.482211	-1.244268	0.737138
18	596	177.804111	33.551991	55.000000	204.5	191.000000	149.500000	1125.736096	-1.540456	0.718296
19	539	134.953618	34.500819	81.000000	235.0	124.666667	154.000000	1190.306519	1.382910	1.500282
20	558	119.942354	5.433460	58.000000	128.0	119.666667	70.000000	29.522490	-4.187761	41.549337
21	590	83.997316	17.278968	37.000000	111.0	79.166667	74.000000	298.562739	0.042042	-1.162589
22	534	116.577247	2.901365	80.500000	121.0	116.000000	40.500000	8.417917	-4.514462	50.124723

Figure 31: Living Room signal Features

5.1.5 Outside

Table 11 shows figures 32 and 33 representing the Outside environment. This environment is different to the other indoor locations, in that it has no shelter from direct sunlight or atmospheric interference. The device is situated away from any obstructions or buildings that could interfere with the ALS reading. The possibility exists for the device to be covered during the later hours of the day by the westerly located house. However, the fact that the recording was performed at midday this risk is negated.

Table 11: Outside Environment

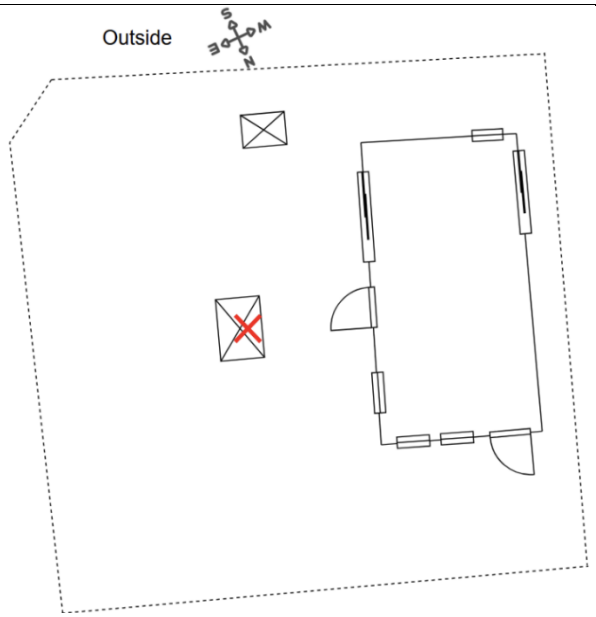

Figure 32: Outside Layout

Figure 33: Outside

Figure 34 represents the raw Outside ALS data recordings. Whilst it shares some similarities with the other indoor data sets, in that some datasets are defined by either high or low volatility, the lux readings are in the thousands. This is directly due to the lack of cover that the ALS recording device has when situated outside. Atmospheric phenomena such as clouds can also be seen in the data through dips in lux values. Whilst the main reason for this is likely to be clouds there could be other phenomena such as objects covering the device. It is also of note that recordings have taken place whilst there has been rain, therefore which could also cause volatility, further limitations and reasonings as to why the data is as it is are outlined in section 3.5.2.

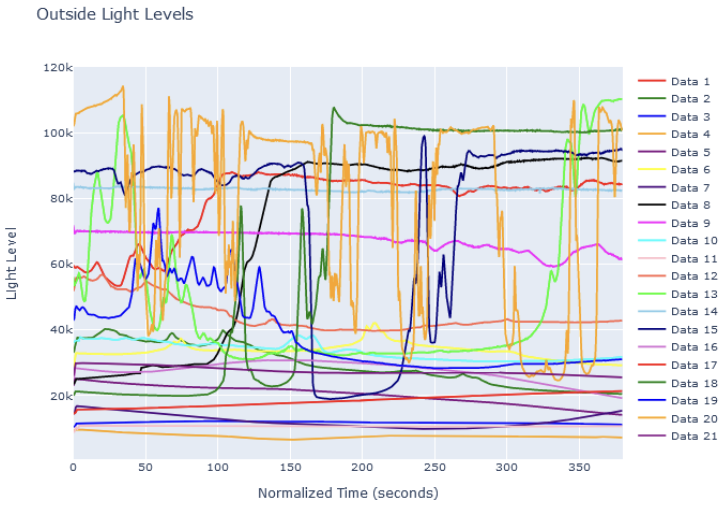

Figure 34: Outside Measured Light Levels

5.1.5.1 Outside Digital representation

Figure 35 shows the extracted key signal features of the Outside ALS dataset. It consists of 21 different data sets, with each dataset having the key features extrapolated. For example, the figure shows the number of data points, the averages of the data, its minimum and maximum and therefore its range as well as statistical descriptions of the data through kurtosis and skewness of the data. Compared to the other indoor datasets the resolution is much higher as seen by the increased number of data points. This is due to direct sunlight interacting with the device. As the ALS device uses a non-uniform recording structure and only records on sensor change, a reason for this could be due to the fact that as the sunlight enters the earth's atmosphere it undergoes atmospheric scattering where photos are scattered by particles in the atmosphere leading to a non-stable stream of photons. Causing more data points to be generated.

Extracted Signal Processing Features:

dataset_index	num_data_points	mean_light_level	std_light_level	min_light_level	max_light_level	median_light_level	light_level_range	light_level_variance	light_level_skewness	light_level_kurtosis
0	3099	79771.987093	9832.342381	53306.500000	88057.000000	84034.000000	34750.500000	9.667496e+07	-1.549998	0.800012
1	3024	28620.373402	6560.509650	18400.000000	40256.500000	27417.250000	21856.500000	4.304029e+07	0.242795	-1.364878
2	2777	11810.251500	331.400110	10169.000000	12260.000000	11758.333333	2091.000000	1.098260e+05	-1.154459	3.641061
3	2733	7694.545402	671.630878	6588.000000	9711.500000	7665.166667	3123.500000	4.510880e+05	0.959685	1.180985
4	3195	19191.611242	3666.387809	10982.000000	25161.000000	19942.000000	14179.000000	1.344240e+07	-0.387072	-1.099620
5	3114	33046.434677	2753.173562	26036.000000	42093.000000	33516.903846	16057.000000	7.579965e+06	0.223494	0.525982
6	2822	12370.810241	2058.759358	9843.000000	16806.000000	11810.500000	6963.000000	4.238490e+06	0.539536	-0.971660
7	2792	70907.715288	27783.222852	23616.500000	92422.000000	89200.000000	68805.500000	7.719075e+08	-0.813616	-1.245030
8	2879	67117.980254	2921.523947	59236.333333	71808.000000	68356.333333	12571.666667	8.535302e+06	-0.906981	-0.179250
9	3060	33920.096432	3500.150192	30354.500000	49852.666667	33448.083333	19498.166667	1.225105e+07	1.529044	3.703343
10	2823	10652.202621	137.993478	8705.000000	10826.000000	10654.368421	2121.000000	1.904220e+04	-9.335850	109.324313
11	3007	44296.161955	4692.006352	39686.000000	56926.000000	42431.800000	17240.000000	2.201492e+07	1.310570	0.427400
12	3145	55069.209279	28828.502198	30974.000000	110808.000000	37220.750000	79834.000000	8.310825e+08	0.971942	-0.697902
13	2857	82735.546027	453.947868	81156.000000	84770.000000	82778.500000	3614.000000	2.060687e+05	-0.194938	-0.533685
14	2983	76732.637513	26608.805345	18959.500000	98981.500000	88845.000000	80022.000000	7.080285e+08	-1.417626	0.263920
15	3066	26380.022043	4183.898772	15539.000000	30888.000000	27636.964286	15349.000000	1.750501e+07	-1.006283	-0.187815
16	3194	18827.385436	1871.280494	14385.000000	21487.000000	18956.062500	7102.000000	3.501691e+06	-0.206707	-1.335798
17	3059	70776.724828	37219.710414	19933.000000	107706.000000	100471.000000	87773.000000	1.385307e+09	-0.466818	-1.698091
18	3045	37983.067323	11383.667022	28199.000000	77158.666667	31538.833333	48959.666667	1.295879e+08	1.133727	0.125442
19	3086	81194.190176	27913.378069	24614.000000	114070.500000	97123.000000	89456.500000	7.791567e+08	-0.849409	-0.799168
20	2854	27445.329625	1241.627916	22822.500000	29867.000000	26962.282258	7044.500000	1.541640e+06	0.534928	-0.588533

Figure 35: Outside Digital Representation

5.2 Classification Models

This section outlines the various machine learning classification models that were applied to the ALS dataset, with a guise to determine the effectiveness of various classification models in differentiating between the different locations. The particularities of each model are explained below and how successful they are in identifying the unique room fingerprints.

5.2.1 J48

Training Set with outside data:

```
Results for J48:

Correctly Classified Instances       62782               99.9395 %
Incorrectly Classified Instances        38                0.0605 %
Kappa statistic                          0.999
Mean absolute error                      0.0003
Root mean squared error                  0.0154
Relative absolute error                  0.1166 %
Root relative squared error              4.3041 %
Total Number of Instances            62820

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.000    0.998      1.000    0.999      0.999    1.000     0.998     Kitchen
                 0.998    0.000    0.999      0.998    0.999      0.998    0.999     0.998     Bedroom
                 0.998    0.000    0.999      0.998    0.999      0.998    0.999     0.998     Landing
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Outside
                 0.999    0.000    0.999      0.999    0.999      0.998    0.999     0.998     LivingRoom
Weighted Avg.    0.999    0.000    0.999      0.999    0.999      0.999    1.000     0.999

=== Confusion Matrix ===

     a     b     c     d     e   <-- classified as
  8558     0     1     0     3 |     a = Kitchen
     9  8064     5     0     3 |     b = Bedroom
     3     3  5349     1     2 |     c = Landing
     0     0     0 35131     0 |     d = Outside
     3     4     1     0  5680 |     e = LivingRoom
```

Figure 36: J48 Outside Training Set

Test set with outside data:

```
Results for J48:

Correctly Classified Instances       22076               59.5153 %
Incorrectly Classified Instances     15017               40.4847 %
Kappa statistic                          0.421
Mean absolute error                      0.1619
Root mean squared error                  0.4023
Relative absolute error                 57.837  %
Root relative squared error            107.5316 %
Total Number of Instances            37093

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.250    0.133    0.222      0.250    0.235      0.111    0.563     0.168     Kitchen
                 0.046    0.085    0.070      0.046    0.056      -0.047   0.528     0.126     Bedroom
                 0.463    0.075    0.456      0.463    0.459      0.386    0.686     0.276     Landing
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Outside
                 0.132    0.174    0.113      0.132    0.122      -0.039   0.481     0.143     LivingRoom
Weighted Avg.    0.595    0.062    0.591      0.595    0.593      0.532    0.772     0.574

=== Confusion Matrix ===

     a     b     c     d     e   <-- classified as
  1220   424   682     0  2562 |     a = Kitchen
  1831   209   652     0  1840 |     b = Bedroom
    24  1245  2047     0  1106 |     c = Landing
     0     0     0 17895     0 |     d = Outside
  2431  1111  1109     0   705 |     e = LivingRoom
```

Figure 37: J48 Outside test set

Training Set without outside data:

```
Evaluating J48...
Results for J48:

Correctly Classified Instances       27645               99.8411 %
Incorrectly Classified Instances        44                0.1589 %
Kappa statistic                          0.9979
Mean absolute error                      0.0009
Root mean squared error                  0.0281
Relative absolute error                  0.2518 %
Root relative squared error              6.5323 %
Total Number of Instances            27689

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.999    0.001    0.998      0.999    0.999      0.998    0.999     0.998     Kitchen
                 0.998    0.001    0.999      0.998    0.998      0.997    0.999     0.998     Bedroom
                 0.999    0.000    0.998      0.999    0.998      0.998    0.999     0.997     Landing
                 0.999    0.000    0.998      0.999    0.999      0.998    0.999     0.997     LivingRoom
Weighted Avg.    0.998    0.001    0.998      0.998    0.998      0.998    0.999     0.998

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 8554    5    1    2 |    a = Kitchen
   10 8061    5    5 |    b = Bedroom
    3    3 5350    2 |    c = Landing
    3    2    3 5680 |    d = LivingRoom
```

Figure 38: J48 Outside Omitted Training set

Test set without outside data:

```
Evaluating J48...
Results for J48:

Correctly Classified Instances        4152               21.6273 %
Incorrectly Classified Instances     15046               78.3727 %
Kappa statistic                         -0.0453
Mean absolute error                      0.3961
Root mean squared error                  0.6264
Relative absolute error                105.8263 %
Root relative squared error            144.7959 %
Total Number of Instances            19198

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.099    0.170    0.166      0.099    0.124      -0.086   0.445     0.256     Kitchen
                 0.188    0.322    0.152      0.188    0.168      -0.126   0.457     0.216     Bedroom
                 0.463    0.197    0.413      0.463    0.437      0.256    0.611     0.288     Landing
                 0.144    0.360    0.134      0.144    0.139      -0.211   0.395     0.260     LivingRoom
Weighted Avg.    0.216    0.265    0.211      0.216    0.211      -0.052   0.472     0.255

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  484 1160 1210 2034 |    a = Kitchen
 1190  850  652 1840 |    b = Bedroom
    9 1260 2047 1106 |    c = Landing
 1233 2308 1044  771 |    d = LivingRoom
```

Figure 39: J48 Outside Omitted Test Set

5.2.2 Naïve Bayes

Training set with outside

```
Evaluating NaiveBayes...
Results for NaiveBayes:

Correctly Classified Instances       48237               76.7861 %
Incorrectly Classified Instances     14583               23.2139 %
Kappa statistic                          0.6341
Mean absolute error                      0.1135
Root mean squared error                  0.241
Relative absolute error                 44.5735 %
Root relative squared error             67.5487 %
Total Number of Instances            62820

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.719    0.117    0.493      0.719    0.585      0.518    0.903     0.501     Kitchen
                 0.283    0.069    0.378      0.283    0.324      0.243    0.819     0.313     Bedroom
                 0.415    0.012    0.766      0.415    0.538      0.536    0.902     0.579     Landing
                 1.000    0.002    0.999      1.000    0.999      0.999    1.000     1.000     Outside
                 0.429    0.066    0.394      0.429    0.411      0.350    0.895     0.400     LivingRoom
Weighted Avg.    0.768    0.033    0.775      0.768    0.763      0.738    0.946     0.753

=== Confusion Matrix ===

     a     b     c     d     e   <-- classified as
  6158  1505     0     0   899 |     a = Kitchen
  2992  2289   653     0  2147 |     b = Bedroom
  1493   899  2221    42   703 |     c = Landing
     0     0     0 35131     0 |     d = Outside
  1855  1369    26     0  2438 |     e = LivingRoom
```

Figure 40: Naïve Bayes Outside Training set

Test set with outside:

```
Evaluating NaiveBayes...
Results for NaiveBayes:

Correctly Classified Instances       23277               62.7531 %
Incorrectly Classified Instances     13816               37.2469 %
Kappa statistic                          0.4675
Mean absolute error                      0.1489
Root mean squared error                  0.2862
Relative absolute error                 53.1805 %
Root relative squared error             76.5037 %
Total Number of Instances            37093

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.306    0.106    0.305      0.306    0.305      0.200    0.814     0.276     Kitchen
                 0.346    0.146    0.248      0.346    0.289      0.174    0.761     0.214     Bedroom
                 0.296    0.027    0.595      0.296    0.396      0.369    0.932     0.568     Landing
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Outside
                 0.188    0.150    0.175      0.188    0.181      0.037    0.728     0.230     LivingRoom
Weighted Avg.    0.628    0.057    0.649      0.628    0.631      0.579    0.899     0.646

=== Confusion Matrix ===

     a     b     c     d     e   <-- classified as
  1495   833     0     0  2560 |     a = Kitchen
  1142  1569    27     0  1794 |     b = Bedroom
     3  2708  1311     0   400 |     c = Landing
     0     0     0 17895     0 |     d = Outside
  2264  1220   865     0  1007 |     e = LivingRoom
```

Figure 41: Naïve Bayes Outside Test set

Training Set without Outside data:

```
Evaluating NaiveBayes...
Results for NaiveBayes:

Correctly Classified Instances       13180               47.6001 %
Incorrectly Classified Instances     14509               52.3999 %
Kappa statistic                          0.2824
Mean absolute error                      0.3207
Root mean squared error                  0.4052
Relative absolute error                 86.7334 %
Root relative squared error             94.2353 %
Total Number of Instances            27689

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.721    0.332    0.493      0.721    0.585      0.361    0.726     0.503     Kitchen
                 0.280    0.190    0.378      0.280    0.322      0.100    0.496     0.312     Bedroom
                 0.424    0.030    0.770      0.424    0.547      0.504    0.750     0.594     Landing
                 0.435    0.171    0.397      0.435    0.415      0.256    0.729     0.402     LivingRoom
Weighted Avg.    0.476    0.199    0.493      0.476    0.466      0.291    0.664     0.444

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 6171 1490    0  901 |    a = Kitchen
 3015 2262  653 2151 |    b = Bedroom
 1505  873 2271  709 |    c = Landing
 1830 1356   26 2476 |    d = LivingRoom
```

Figure 42: Naive Bayes Outside Omitted Training Set

Test Set without Outside data:

```
Evaluating NaiveBayes...
Results for NaiveBayes:

Correctly Classified Instances        5871               30.5813 %
Incorrectly Classified Instances     13327               69.4187 %
Kappa statistic                          0.0705
Mean absolute error                      0.3595
Root mean squared error                  0.4449
Relative absolute error                 96.0481 %
Root relative squared error            102.8539 %
Total Number of Instances            19198

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.307    0.210    0.333      0.307    0.320      0.100    0.583     0.276     Kitchen
                 0.369    0.329    0.258      0.369    0.304      0.036    0.468     0.215     Bedroom
                 0.296    0.060    0.596      0.296    0.396      0.312    0.850     0.570     Landing
                 0.259    0.334    0.231      0.259    0.244      -0.072   0.375     0.228     LivingRoom
Weighted Avg.    0.306    0.238    0.347      0.306    0.312      0.086    0.560     0.316

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1502  856    0 2530 |    a = Kitchen
 1142 1673   26 1691 |    b = Bedroom
    3 2713 1310  396 |    c = Landing
 1858 1250  862 1386 |    d = LivingRoom
```

Figure 43: Naive Bayes Outside Omitted Test Set

5.2.3 Logistic Regression

Training set with outside data:

```
Evaluating Logistic...
Results for Logistic:

Correctly Classified Instances       48644               77.4339 %
Incorrectly Classified Instances     14176               22.5661 %
Kappa statistic                          0.6444
Mean absolute error                      0.1144
Root mean squared error                  0.239
Relative absolute error                 44.921  %
Root relative squared error             66.9738 %
Total Number of Instances            62820

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.671    0.119    0.471      0.671    0.553      0.479    0.902     0.508     Kitchen
               0.359    0.072    0.425      0.359    0.390      0.310    0.848     0.335     Bedroom
               0.521    0.022    0.690      0.521    0.594      0.568    0.921     0.643     Landing
               1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Outside
               0.365    0.045    0.449      0.365    0.403      0.352    0.891     0.341     LivingRoom
Weighted Avg.  0.774    0.031    0.778      0.774    0.772      0.745    0.950     0.757

=== Confusion Matrix ===

     a     b     c     d     e   <-- classified as
  5745  1555   348     0   914 |     a = Kitchen
  3286  2904   861     0  1030 |     b = Bedroom
  1305   664  2790     0   599 |     c = Landing
     0     0     0 35131     0 |     d = Outside
  1870  1702    42     0  2074 |     e = LivingRoom
```

Figure 44: Logistic Regression Outside Training Set

Test set with outside data:

```
Evaluating Logistic...
Results for Logistic:

Correctly Classified Instances       23098               62.2705 %
Incorrectly Classified Instances     13995               37.7295 %
Kappa statistic                          0.4611
Mean absolute error                      0.1462
Root mean squared error                  0.2766
Relative absolute error                 52.207  %
Root relative squared error             73.9249 %
Total Number of Instances            37093

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.106    0.065    0.199      0.106    0.138      0.055    0.831     0.294     Kitchen
               0.375    0.138    0.274      0.375    0.317      0.208    0.807     0.285     Bedroom
               0.442    0.073    0.452      0.442    0.447      0.373    0.934     0.572     Landing
               1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Outside
               0.192    0.159    0.170      0.192    0.180      0.032    0.729     0.219     LivingRoom
Weighted Avg.  0.623    0.057    0.621      0.623    0.619      0.564    0.907     0.656

=== Confusion Matrix ===

     a     b     c     d     e   <-- classified as
   518  1371   311     0  2688 |     a = Kitchen
   455  1698   398     0  1981 |     b = Bedroom
     0  2097  1956     0   369 |     c = Landing
     0     0     0 17895     0 |     d = Outside
  1631  1031  1663     0  1031 |     e = LivingRoom
```

Figure 45: Logistic Regression Outside Test Set

Training set without outside data:

```
Evaluating Logistic...
Results for Logistic:

Correctly Classified Instances       13511               48.7956 %
Incorrectly Classified Instances     14178               51.2044 %
Kappa statistic                          0.2979
Mean absolute error                      0.3244
Root mean squared error                  0.4025
Relative absolute error                 87.7392 %
Root relative squared error             93.5994 %
Total Number of Instances            27689

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.670    0.338    0.470      0.670    0.553      0.309    0.721     0.508     Kitchen
                 0.360    0.200    0.425      0.360    0.390      0.168    0.576     0.335     Bedroom
                 0.522    0.056    0.693      0.522    0.596      0.522    0.796     0.643     Landing
                 0.364    0.116    0.448      0.364    0.401      0.269    0.717     0.341     LivingRoom
Weighted Avg.    0.488    0.197    0.496      0.488    0.482      0.301    0.692     0.449

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 5738 1560  346  918 |    a = Kitchen
 3291 2906  854 1030 |    b = Bedroom
 1301  659 2799  599 |    c = Landing
 1869 1709   42 2068 |    d = LivingRoom
```

Figure 46: Logistic Regression Outside Omitted Training Set

Test set without outside data:

```
Evaluating Logistic...
Results for Logistic:

Correctly Classified Instances        5207               27.1226 %
Incorrectly Classified Instances     13991               72.8774 %
Kappa statistic                          0.0272
Mean absolute error                      0.353
Root mean squared error                  0.4298
Relative absolute error                 94.3158 %
Root relative squared error             99.3607 %
Total Number of Instances            19198

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.106    0.146    0.199      0.106    0.138      -0.051   0.621     0.294     Kitchen
                 0.375    0.306    0.274      0.375    0.317      0.062    0.571     0.285     Bedroom
                 0.443    0.161    0.452      0.443    0.448      0.285    0.853     0.572     Landing
                 0.192    0.364    0.170      0.192    0.180      -0.165   0.378     0.219     LivingRoom
Weighted Avg.    0.271    0.248    0.267      0.271    0.263      0.021    0.595     0.335

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  518 1371  311 2688 |    a = Kitchen
  455 1698  398 1981 |    b = Bedroom
    0 2093 1960  369 |    c = Landing
 1631 1031 1663 1031 |    d = LivingRoom
```

Figure 47: Logistic Regression Outside Omitted Test Set

5.2.4 Other Classification Models

As well as the classification algorithms, J48, Naïve Bayes and Logistic Regression, a number of other classification models were attempted. Table 12 shows the rest of the ones that were experimented on. It is also of note that certain classification models were attempted which were completely unsuited, these being, Multilayer Perceptron, AdaBoostM1, ZeroR, OneR, and Decision Table.

Table 12: Machine Learning Classification Results

Learning Algorithm Name	Results	Results (Outside omitted)
J48	59	21
Naive Bayes	62	30
SMO	57	26
IBk	50	13
Random Forest	53	12
Logistic Regression	62	27
Multilayer Perceptron (Error)	58	20
RepTree	53	12
BaynesNet	49	8
AdaBoostM1(Error)	61	32
Bagging	53	12
Stacking	48	25
ZeroR(Error)	58	25
OneR	0	3
PART	58	18
Decision Table	0	3

It is also important to note that some classification models were tested that were completely unsuited to the task. As shown in Table 12

Chapter 6: Discussion

In this section, the research aims and objectives will be systematically addressed and discussed. The discussion will outline and evaluate how each objective was met, the significance of the findings found in Chapter 5, and explore the implications of the results in the context of the existing research literature around localisation with light sensors. furthermore, this chapter will outline and acknowledge the limitations of the dissertation and make research recommendations for future work.

6.1 Research Objectives and Key Findings

Referring to the research aims and objectives specified in section 1.1.1, this discussion section will evaluate how each point has been answered and how the significance behind each objective to the context of the research question "*How can smartphone ambient light sensors (ALS) be exploited to infer environmental information and what measures can be taken to mitigate the potential privacy and security risks?"*

The first objective, to conduct a literature review to understand the current state of research surrounding ambient light sensors and the security and privacy concerns, along with the implications of PII environmental inference and personal security has been met as the literature review in 2.6 provides a solid foundation for the research conducted on this dissertation. Light-based related works highlighted the issues with innate Android security systems and the potential for information inference attacks on a user. The literature review also allowed for gaps in the existing research base to be identified and for this dissertation to be put into context with the existing light literature.

The second objective, to design and develop an Android application to collect the smartphone's ALS data has been met as this dissertation demonstrated in section 4.2 that an application recording ambient light from a smartphone's ALS is possible. The significance of this is that it highlights that this readily available data can be collected without a user's express consent or knowledge. Which could be used potentially by a malicious actor for nefarious purposes.

The third objective, to utilise Google Colab to present and format captured ambient light sensor data has been met, as shown in Chapter 5, the environmental lux visualisations enable a clear visual overview of each environment. The significance of this data visualisation is that it allows for preliminary data analysis by identifying trends and anomalies to show the distinctiveness of each environment. For example, through the use of visual representations, inside and outside locations can be determined, thus demonstrating that at a basic level, with basic data visualisation location inference is possible.

The fourth objective, to test in different common household locations, such as rooms and outside environments was met as in section 5.1, five locations, four inside and one outside location were presented. This provided a basis to test out the case presented by the research question. The outside location was chosen to allow for the comparison of indoor and outdoor paradigms.

The fifth objective, to investigate the risk around how a malicious actor would be able to exploit the vulnerabilities of ALS to violate PII was met in section 2.6. Where related research highlights that while to some degree having access to the ALS data could give a malicious actor access to PII, the real risk comes from the combination of data from multiple sensors. Such as accelerometers and gyroscopes for indoor localisation. These findings were significant as they show the risk and impact of seemingly innocuous information.

The sixth objective is to assess and evaluate how effective the security measures surrounding mobile ambient light sensors are. This objective was met in Chapter 2, where the Android security model is

outlined and how certain elements such as access control and permission settings provide a baseline security level. However, this has been found to be inadequate to mitigate the risks associated with ALS data. The significance of this is that the research highlights the need for more robust security practices that address indirect information leakage attacks such as those described in section 2.2.1.

The seventh objective, to propose possible solutions and risk mitigation strategies to address any ineffective ambient light sensor security measures was partially met in section 2.4. This dissertation described common attacks that ALS are susceptible to and provided mitigation strategies and solutions, including the implementation of stronger Android security frameworks through granular access controls. The significance of this is that while mitigation strategies do exist that have not been employed in the standard Android stack, possibly due to their pros and cons. Third-party developers such as OEMs are able to implement these described systems to mitigate these risks.

6.2 Limitations

While this dissertation has been successful in answering and achieving the stated research objectives and has provided insight into locational inference of ALS, there are several overarching limitations that must be addressed, that impact the certainty and quality of the answer to the research question. These constraints stem from the aforementioned limitations discussed in section 3.5.2.

The short time span of the sample data at seven minutes per day per environment is insufficient to provide enough differentiation between environmental datasets and account for environmental interference. This highlights the fact that the results might differ depending on the time of day, or time of year, which should be accounted for. Furthermore, as only one device was used for data collection this project does not attempt to account for hardware irregularities that could give differing results.

As each environment was set up to not include any artificial lighting and the location of the smartphone device was positioned so that it would be the most optimal in terms of light, this does not reflect real-world scenarios. For example, the devices were placed stationary where in a real-world situation the device could be moving around, or placed in direct sunlight.

The android sensor manager API only exposes the lux value, even on RGB-enabled devices, meaning that the reduced dimensionality of the data would affect a dataset's 'uniqueness'. It also removes the avenue for more advanced analysis techniques such as colour spectroscopy.

While machine learning was tested on the dataset, the limited dimensionality and sample size increased the risk of overfitting giving less than ideal results. one of the reasons that the ML was poor is that there was a lack of variability in the data hence features could not be identified. This lack of variability can be measured by calculating the entropy [245] in a sample. If the entropy is too low it is unlikely that a feature that can identify the case can be found. Calculating Entropy and using an entropy threshold could be used as a test to see if sufficient data has been collected in a similar manner to the data saturation concept [246]in qualitative analysis. Figure 55 shows preliminary work into the entropy of the ALS dataset to highlight a cause for the disparity between the inside and outside environments.

This project also could have explored machine learning elements such as hyperparameters to fine-tune classification models to give better results.

Overall, these limitations hinder the effectiveness of the results that answer the research question. While at a basic level of differentiating between indoors and outdoors is possible at this level of analysis further work is needed for inter-room classification.

6.3 Summary

In summary, the recommendations for future work that have been outlined above provide a substantial response to the limitations present within this dissertation. By making data collection periods longer, increasing dimensionality within the ALS data, and applying well-configured machine learning classification algorithms can help broaden the general understanding of ALS data in mobile devices. Whilst contributing to a more security and privacy-aware Android sensor environment, meaning that users would be able to enjoy the full functionalities of their devices without compromising PII, and if not possible then give users a better understanding of the risks associated with location inference.

Chapter 7: Conclusion

In this chapter, the research question "*How can smartphone ambient light sensors (ALS) be exploited to infer environmental information and what measures can be taken to mitigate the potential privacy and security risks"* is answered.

Based on the results in Chapter 5 and the analysis of the findings that have been conducted throughout this dissertation. Work in this project has shown that at least to some degree ALS data can be used to perform inference attacks to perform environmental localisation and in turn violate the user's PII. As the data from the ALS system is usually considered innocuous, android allows for the light sensor to be permissionless. Meaning inference attacks can be carried out without the user's knowledge.

However, while it has been shown that differentiating between indoor and outdoor environments is easily achieved, this project's limitations of short data collection periods, device-specific hardware variations, and controlled environments affect the results in a real-world situation. Especially when performing analysis against multiple indoor environments due to their homogeneity.

The analysis performed in this dissertation has also highlighted gaps in the current security measures that surround ALS data. While the Android Security model provides protections, they are not sufficient in mitigating cyber-attacks, inference attacks especially. As a response, this project recommends some mitigation strategies and frameworks that could be implemented to reduce the overall Android attack surface. For example, implementing more granular access-based controls within the Android ecosystem as a whole.

7.1 Future work

Based on the results and the limitations present within this dissertation, there are several recommendations that can be made for future works.

7.1.1 Extended recording periods and Environments

As outlined in section 6.2, one of the issues present within this project is that the seven-minute data collection period is insufficient to remove the influence of external temporary anomalies. While these anomalies technically do contribute to the ambient light of an environment, phenomena such as cloud cover can occur well beyond the seven-minute recording period. Meaning that it could be possible that for that time period, the ambient light was not an accurate representation of the actual ambient light of the room. Any future work should consider recording over extended periods of time to encompass different times of the day, different seasons and varying weather conditions. Having an extended body of data to collect from would help negate the influence of anomalous environmental phenomena.

Another point to consider is that while the chosen environments represent a domestic setting, future work could encompass other real-world environments such as public transport, commercial buildings and other outdoor environments such as woods or forests. By taking this more complex approach a clearer understanding of the impacts and limitations of locational inference would become clearer.

7.1.2 Multi-dimensional data

Just as previous work mentioned in section 2.6, shows that the combinations of different sensors create a wider attack surface in which more useful information could be utilised. Another related potential is that of RGB ALS, in which red green and blue sensors could be used to provide increased dimensionality to the dataset. However, as outlined in section 3.5.2, android security policies reduce functionality via the SensorManager API. The possibility for distinguishing between different types of light also becomes possible depending on individual device specifications.

7.1.3 Expanded Machine Learning Analysis

Another possible avenue for improvement or continuation is that of the data analysis. As described in section 6.2, preliminary data analysis was performed using the aforementioned classification algorithms on the ASL datasets. While this basic research provided some valuable insights, there does remain a significant potential for future work surrounding machine learning analysis.

Advanced machine learning techniques and practises should be applied to the ALS datasets in order to gain more insight and achieve a higher general positive accuracy score on the classification models. As mentioned before one of these avenues could be through the use of hyperparameter optimisation or through ensemble learning approaches. Another route for future work in machine learning classification could be the use of deep learning techniques to employ neural networks to perform data classification to a higher level of accuracy. However, these techniques often require larger sample sizes, which is an issue this project suffers from.

REFERENCES

[1] ‘Ambient Light Sensors - STMicroelectronics’. Accessed: Aug. 27, 2024. [Online]. Available: https://www.st.com/en/imaging-and-photonics-solutions/ambient-light-sensors.html

[2] ‘Google Scholar’. Accessed: Aug. 24, 2024. [Online]. Available: https://scholar.google.com/

[3] ‘ResearchGate | Find and share research’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.researchgate.net/

[4] ‘Scopus preview - Scopus - Welcome to Scopus’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.scopus.com/home.uri

[5] ‘Royal Holloway LibrarySearch’. Accessed: Aug. 24, 2024. [Online]. Available: https://librarysearch.royalholloway.ac.uk

[6] ‘Computer and Internet Use in the United States: 2016’.

[7] ‘IoT connections worldwide 2022-2033’, Statista. Accessed: Aug. 24, 2024. [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[8] D. Chaffey, ‘Global social media statistics research summary 2024 [May 2024]’, Smart Insights. Accessed: Aug. 24, 2024. [Online]. Available: https://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/

[9] A. C. Davison, Y. Dodge, and N. Wermuth, *Celebrating Statistics*. Oxford University Press, 2005. doi: 10.1093/acprof:oso/9780198566540.001.0001.

[10] ‘Great Internet Mersenne Prime Search - PrimeNet’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.mersenne.org/

[11] S. S. Gill *et al.*, ‘Modern computing: Vision and challenges’, *Telemat. Inform. Rep.*, vol. 13, p. 100116, Mar. 2024, doi: 10.1016/j.teler.2024.100116.

[12] ‘ENIAC | History, Computer, Stands For, Machine, & Facts | Britannica’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.britannica.com/technology/ENIAC

[13] ‘What Is a Mainframe? | IBM’. Accessed: Aug. 26, 2024. [Online]. Available: https://www.ibm.com/topics/mainframe

[14] ‘What Is a Data Center?’, Cisco. Accessed: Aug. 26, 2024. [Online]. Available: https://www.cisco.com/c/en_uk/solutions/data-center-virtualization/what-is-a-data-center.html

[15] ‘ISO 27001:2013 – Annex A.11: Physical & Environmental Security | ISMS.online’, https://www.isms.online/. Accessed: Aug. 24, 2024. [Online]. Available: https://www.isms.online/iso-27001/annex-a-11-physical-and-environmental-security/

[16] jenna, ‘Miniaturisation Of Electronics: The Complete Guide’, MPE Electronics. Accessed: Aug. 24, 2024. [Online]. Available: https://www.mpe-electronics.co.uk/2024/03/12/miniaturisation-of-electronics-the-complete-guide

[17] ‘What is a personal digital assistant (PDA)?’, Mobile Computing. Accessed: Aug. 24, 2024. [Online]. Available: https://www.techtarget.com/searchmobilecomputing/definition/personal-digital-assistant

[18] ‘What is PDA & How does a PDA differ from a smartphone? | Lenovo IN’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.lenovo.com/in/en/glossary/pda/

[19] ‘IBM - Simon’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.mobilephonemuseum.com/phone-detail/ibm-simon

[20] ‘What Is Cloud Computing? | Microsoft Azure’. Accessed: Aug. 26, 2024. [Online]. Available: https://azure.microsoft.com/en-gb/resources/cloud-computing-dictionary/what-is-cloud-computing

[21] ‘What is SaaS? Software as a Service | Microsoft Azure’. Accessed: Aug. 24, 2024. [Online]. Available: https://azure.microsoft.com/en-gb/resources/cloud-computing-dictionary/what-is-saas

[22] ‘What is PaaS? Platform as a Service | Microsoft Azure’. Accessed: Aug. 24, 2024. [Online]. Available: https://azure.microsoft.com/en-gb/resources/cloud-computing-dictionary/what-is-paas

[23] ‘What is IaaS? Infrastructure as a Service | Microsoft Azure’. Accessed: Aug. 24, 2024. [Online]. Available: https://azure.microsoft.com/en-gb/resources/cloud-computing-dictionary/what-is-iaas
[24] I. Chowdhury and M. Zulkernine, ‘Using complexity, coupling, and cohesion metrics as early indicators of vulnerabilities’, *J. Syst. Archit.*, vol. 57, no. 3, pp. 294–313, Mar. 2011, doi: 10.1016/j.sysarc.2010.06.003.
[25] F. Camilo, A. Meneely, and M. Nagappan, ‘Do Bugs Foreshadow Vulnerabilities? A Study of the Chromium Project’, in *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, May 2015, pp. 269–279. doi: 10.1109/MSR.2015.32.
[26] A. A. Press, ‘Melbourne student health records posted online in “appalling” privacy breach’, *The Guardian*, Aug. 22, 2018. Accessed: Aug. 24, 2024. [Online]. Available: https://www.theguardian.com/australia-news/2018/aug/22/melbourne-student-health-records-posted-online-in-appalling-privacy-breach
[27] ‘What Is a Computer Worm and How Does It Work?’, Security. Accessed: Aug. 24, 2024. [Online]. Available: https://www.techtarget.com/searchsecurity/definition/worm
[28] ‘What Is a Legacy System and What Are Legacy Applications? | Definition from TechTarget.com’, IT Operations. Accessed: Aug. 26, 2024. [Online]. Available: https://www.techtarget.com/searchitoperations/definition/legacy-application
[29] S. Al-Rabiaah, ‘The “Stuxnet” Virus of 2010 As an Example of A “APT” and Its “Recent” Variances’, in *2018 21st Saudi Computer Society National Computer Conference (NCC)*, Apr. 2018, pp. 1–5. doi: 10.1109/NCG.2018.8593143.
[30] ‘What are Insider Threats? | IBM’. Accessed: Aug. 26, 2024. [Online]. Available: https://www.ibm.com/topics/insider-threats
[31] ‘19 Types of Phishing Attacks with Examples’, Fortinet. Accessed: Aug. 24, 2024. [Online]. Available: https://www.fortinet.com/resources/cyberglossary/types-of-phishing-attacks
[32] ‘Weak Security Controls and Practices Routinely Exploited for Initial Access | CISA’. Accessed: Aug. 26, 2024. [Online]. Available: https://www.cisa.gov/news-events/cybersecurity-advisories/aa22-137a
[33] ‘Confidentiality, Integrity, and Availability: The CIA Triad | Office of Information Security | Washington University in St. Louis’. Accessed: Aug. 24, 2024. [Online]. Available: https://informationsecurity.wustl.edu/items/confidentiality-integrity-and-availability-the-cia-triad/
[34] C. C. Editor, ‘vulnerability - Glossary | CSRC’. Accessed: Aug. 24, 2024. [Online]. Available: https://csrc.nist.gov/glossary/term/vulnerability
[35] iamnoahfranklin, ‘The Complex Landscape of Vulnerability Management: Challenges and Solutions’, Medium. Accessed: Aug. 26, 2024. [Online]. Available: https://medium.com/@iamnoahfranklin/the-complex-landscape-of-vulnerability-management-challenges-and-solutions-e5a17544522d
[36] B. R. A. Jr and 2023 Aug 02, ‘Just Why Are So Many Cyber Breaches Due to Human Error? - ’, Security Today. Accessed: Aug. 24, 2024. [Online]. Available: https://securitytoday.com/Articles/2022/07/30/Just-Why-Are-So-Many-Cyber-Breaches-Due-to-Human-Error.aspx
[37] ‘Vulnerability patching 101 | Challenges and Best Practices | SuperOps.ai’. Accessed: Aug. 24, 2024. [Online]. Available: https://superops.com/patch-management/vulnerability-patching
[38] ‘What is cyber security?’ Accessed: Aug. 24, 2024. [Online]. Available: https://www.ncsc.gov.uk/section/about-ncsc/what-is-cyber-security
[39] K. Ahmad, S. Verma, N. Kumar, and J. Shekhar, ‘Classification of Internet Security Attacks’.
[40] ‘What is an Active Attack? - Definition from WhatIs.com’, WhatIs. Accessed: Aug. 24, 2024. [Online]. Available: https://www.techtarget.com/whatis/definition/active-attack
[41] ‘Denial of Service (DoS) guidance’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.ncsc.gov.uk/collection/denial-service-dos-guidance-collection
[42] ‘Understanding Denial-of-Service Attacks | CISA’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.cisa.gov/news-events/news/understanding-denial-service-attacks
[43] ‘SQL Injection | OWASP Foundation’. Accessed: Aug. 24, 2024. [Online]. Available: https://owasp.org/www-community/attacks/SQL_Injection

[44] C. C. Editor, ‘replay attack - Glossary | CSRC’. Accessed: Aug. 24, 2024. [Online]. Available: https://csrc.nist.gov/glossary/term/replay_attack
[45] ‘masquerading - Glossary | CSRC’. Accessed: Aug. 24, 2024. [Online]. Available: https://csrc.nist.gov/glossary/term/masquerading
[46] C. C. Editor, ‘man-in-the-middle attack (MitM) - Glossary | CSRC’. Accessed: Aug. 24, 2024. [Online]. Available: https://csrc.nist.gov/glossary/term/man_in_the_middle_attack
[47] ‘Eavesdropping Attack - an overview | ScienceDirect Topics’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/eavesdropping-attack
[48] ‘Traffic Analysis - an overview | ScienceDirect Topics’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/traffic-analysis
[49] C. C. Editor, ‘Side-Channel Attack - Glossary | CSRC’. Accessed: Aug. 24, 2024. [Online]. Available: https://csrc.nist.gov/glossary/term/side_channel_attack
[50] ‘Figure 2: Passive attacks categories 3.1. Release of Message Content...’, ResearchGate. Accessed: Aug. 24, 2024. [Online]. Available: https://www.researchgate.net/figure/Passive-attacks-categories-31-Release-of-Message-Content-This-attacker-happens-when_fig1_369739760
[51] ‘What is an on-path attacker?’ Accessed: Aug. 24, 2024. [Online]. Available: https://www.cloudflare.com/learning/security/threats/on-path-attack/
[52] Y. Gilad, A. Herzberg, and H. Shulman, ‘Off-Path Hacking: The Illusion of Challenge-Response Authentication’, *IEEE Secur. Priv.*, vol. 12, no. 5, pp. 68–77, Sep. 2014, doi: 10.1109/MSP.2013.130.
[53] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, ‘The Locator/ID Separation Protocol (LISP)’, RFC Editor, RFC6830, Jan. 2013. doi: 10.17487/rfc6830.
[54] ‘Check Point response to “Off-Path TCP Sequence Number Inference Attack”’. Accessed: Aug. 24, 2024. [Online]. Available: https://support.checkpoint.com/results/sk/sk74640
[55] C. Clavier, B. Feix, G. Gagnerot, and M. Roussellet, ‘Passive and Active Combined Attacks on AES Combining Fault Attacks and Side Channel Analysis’, in *2010 Workshop on Fault Diagnosis and Tolerance in Cryptography*, Aug. 2010, pp. 10–19. doi: 10.1109/FDTC.2010.17.
[56] C. Tankard, ‘Advanced Persistent threats and how to monitor and deter them’, *Netw. Secur.*, vol. 2011, no. 8, pp. 16–19, Aug. 2011, doi: 10.1016/S1353-4858(11)70086-1.
[57] ‘A new approach to China’, Official Google Blog. Accessed: Aug. 24, 2024. [Online]. Available: https://googleblog.blogspot.com/2010/01/new-approach-to-china.html
[58] T. O’Connor, ‘Chapter 4 - Network Traffic Analysis with Python’, in *Violent Python*, T. O’Connor, Ed., Syngress, 2013, pp. 125–169. doi: 10.1016/B978-1-59-749957-6.00004-1.
[59] DJ, ‘Why You Need a Cyber Security Strategy For Your Business’, Intrix Cyber Security. Accessed: Aug. 26, 2024. [Online]. Available: https://intrix.com.au/articles/why-you-need-a-cyber-security-strategy-in-your-business/
[60] otw, ‘The Hacker Methodology’, hackers-arise. Accessed: Aug. 24, 2024. [Online]. Available: https://www.hackers-arise.com/post/the-hacker-methodology
[61] ‘Cyber Kill Chain® | Lockheed Martin’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html
[62] ‘Nmap: the Network Mapper - Free Security Scanner’. Accessed: Aug. 24, 2024. [Online]. Available: https://nmap.org/
[63] ‘What is a RAT (Remote Access Trojan)? | Definition from TechTarget’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.techtarget.com/searchsecurity/definition/RAT-remote-access-Trojan
[64] ‘MITRE ATT&CK®’. Accessed: Aug. 24, 2024. [Online]. Available: https://attack.mitre.org/
[65] ‘Cybersecurity Framework’, *NIST*, Nov. 2013, Accessed: Aug. 24, 2024. [Online]. Available: https://www.nist.gov/cyberframework
[66] N. Naik, P. Jenkins, P. Grace, and J. Song, ‘Comparing Attack Models for IT Systems: Lockheed Martin’s Cyber Kill Chain, MITRE ATT&CK Framework and Diamond Model’, in *2022 IEEE International Symposium on Systems Engineering (ISSE)*, Oct. 2022, pp. 1–7. doi: 10.1109/ISSE54508.2022.10005490.

[67] N. Naik, P. Grace, P. Jenkins, K. Naik, and J. Song, 'An evaluation of potential attack surfaces based on attack tree modelling and risk matrix applied to self-sovereign identity', *Comput. Secur.*, vol. 120, p. 102808, Sep. 2022, doi: 10.1016/j.cose.2022.102808.
[68] S. Ltd, 'MITRE ATT&CK - MDR documentation'. Accessed: Aug. 24, 2024. [Online]. Available: https://docs.sophos.com/central/mdr/help/en-us/welcomeGuides/MDR/mitre/
[69] 'How MITRE ATT&CK Alignment Supercharges Your SIEM', Securonix. Accessed: Aug. 24, 2024. [Online]. Available: https://www.securonix.com/blog/how-mitre-attck-alignment-supercharges-your-siem/
[70] 'What is a Zero-Day Exploit? | IBM'. Accessed: Aug. 24, 2024. [Online]. Available: https://www.ibm.com/topics/zero-day
[71] 'National Institute of Standards and Technology'. Accessed: Aug. 24, 2024. [Online]. Available: https://www.nist.gov/
[72] 'Mobile Computing - an overview | ScienceDirect Topics'. Accessed: Aug. 24, 2024. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/mobile-computing
[73] 'What is a Mobile Phone? - Definition from Techopedia'. Accessed: Aug. 24, 2024. [Online]. Available: https://www.techopedia.com/definition/2955/mobile-phone
[74] 'Portable computer definition – Glossary | NordVPN'. Accessed: Aug. 24, 2024. [Online]. Available: https://nordvpn.com/cybersecurity/glossary/portable-computer/
[75] 'What is wearable computer? | Definition from TechTarget'. Accessed: Aug. 24, 2024. [Online]. Available: https://www.techtarget.com/iotagenda/definition/wearable-computer
[76] 'Charted: There are more mobile phones than people in the world', World Economic Forum. Accessed: Aug. 24, 2024. [Online]. Available: https://www.weforum.org/agenda/2023/04/charted-there-are-more-phones-than-people-in-the-world/
[77] 'Importance of Mobile Phones!' Accessed: Aug. 24, 2024. [Online]. Available: https://www.streetdirectory.com/travel_guide/132870/cell_phones/importance_of_mobile_phones.html
[78] 'Original Equipment Manufacturer (OEM): Definition and Examples'. Accessed: Aug. 24, 2024. [Online]. Available: https://www.investopedia.com/terms/o/oem.asp
[79] 'Android has created more choice, not less', Google. Accessed: Aug. 24, 2024. [Online]. Available: https://blog.google/around-the-globe/google-europe/android-has-created-more-choice-not-less/
[80] 'Android Open Source Project', Android Open Source Project. Accessed: Aug. 24, 2024. [Online]. Available: https://source.android.com/
[81] 'Platform architecture', Android Developers. Accessed: Aug. 24, 2024. [Online]. Available: https://developer.android.com/guide/platform
[82] 'Compatibility Test Suite', Android Open Source Project. Accessed: Aug. 24, 2024. [Online]. Available: https://source.android.com/docs/compatibility/cts
[83] A. Waheed, M. Z. Khan, and A. Vapen, 'Attacks against Smartphones'.
[84] I. Gurulian, R. N. Akram, K. Markantonakis, and K. Mayes, 'Preventing relay attacks in mobile transactions using infrared light', in *Proceedings of the Symposium on Applied Computing*, Marrakech Morocco: ACM, Apr. 2017, pp. 1724–1731. doi: 10.1145/3019612.3019794.
[85] 'An architecture for secure mobile devices', doi: 10.1002/sec.1028.
[86] R. Mayrhofer *et al.*, 'The Android Platform Security Model (2023)', *ACM Trans. Priv. Secur.*, vol. 24, no. 3, pp. 1–35, Aug. 2021, doi: 10.1145/3448609.
[87] 'Android Compatibility Definition Document', Android Open Source Project. Accessed: Aug. 24, 2024. [Online]. Available: https://source.android.com/docs/compatibility/cdd
[88] 'FIGURE 2. Android Software Stack.', ResearchGate. Accessed: Aug. 24, 2024. [Online]. Available: https://www.researchgate.net/figure/Android-Software-Stack_fig2_322515716
[89] 'Application Layer Security - CDNetworks'. Accessed: Aug. 24, 2024. [Online]. Available: https://www.cdnetworks.com/glossary/application-layer-security/
[90] 'Mobile App Security: Ways to Avoid Data Leakage in Android Apps', RipenApps Official Blog For Mobile App Design & Development. Accessed: Aug. 24, 2024. [Online]. Available: https://ripenapps.com/blog/mobile-app-security-ways-to-avoid-data-leakage-in-android-apps/

[91] ‘Lesson 18: Application Isolation — OSU DevOps BootCamp 0.0.1 documentation’. Accessed: Aug. 24, 2024. [Online]. Available: https://devopsbootcamp.osuosl.org/application-isolation.html
[92] ‘Application fundamentals’, Android Developers. Accessed: Aug. 24, 2024. [Online]. Available: https://developer.android.com/guide/components/fundamentals
[93] ‘Notification Manager in Android’, GeeksforGeeks. Accessed: Aug. 24, 2024. [Online]. Available: https://www.geeksforgeeks.org/notification-manager-in-android/
[94] ‘Intent | Android Developers’. Accessed: Aug. 24, 2024. [Online]. Available: https://developer.android.com/reference/android/content/Intent
[95] A. Holmberg, ‘iOS vs Android: Security of Inter-App Communication’.
[96] ‘Swish - About Swish’, Swish. Accessed: Aug. 24, 2024. [Online]. Available: https://www.swish.nu/about-swish
[97] S. Ahmad, ‘Securing Inter Application Connectivity’, Medium. Accessed: Aug. 24, 2024. [Online]. Available: https://medium.com/@thesaadahmad/securing-inter-application-connectivity-44618b6ec28e
[98] ‘Android runtime and Dalvik’, Android Open Source Project. Accessed: Aug. 24, 2024. [Online]. Available: https://source.android.com/docs/core/runtime
[99] IPMU, ‘The MavenGate attack could let hackers to take control of Java and Android systems by exploiting abandoned libraries’, IPMU. Accessed: Aug. 24, 2024. [Online]. Available: https://ipmuonline.com/the-mavengate-attack-could-let-hackers-to-take-control-of-java-and-android-systems-by-exploiting-abandoned-libraries/
[100] ‘Maven – Welcome to Apache Maven’. Accessed: Aug. 24, 2024. [Online]. Available: https://maven.apache.org/
[101] ‘What is a DNS TXT record?’ Accessed: Aug. 27, 2024. [Online]. Available: https://www.cloudflare.com/learning/dns/dns-records/dns-txt-record/
[102] ‘The difference between libraries and frameworks - Simple Talk’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.red-gate.com/simple-talk/featured/the-difference-between-libraries-and-frameworks/
[103] ‘Kernel overview’, Android Open Source Project. Accessed: Aug. 24, 2024. [Online]. Available: https://source.android.com/docs/core/architecture/kernel
[104] M. Linares-Vásquez, G. Bavota, and C. Escobar-Velásquez, ‘An Empirical Study on Android-Related Vulnerabilities’, in *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*, May 2017, pp. 2–13. doi: 10.1109/MSR.2017.60.
[105] ‘Dirty COW (CVE-2016-5195)’. Accessed: Aug. 24, 2024. [Online]. Available: https://dirtycow.ninja/
[106] ‘Kernel Self Protection Project’, Linux Kernel Self-Protection Project. Accessed: Aug. 24, 2024. [Online]. Available: https://kspp.github.io/
[107] ‘z/OS 2.4.0’. Accessed: Aug. 27, 2024. [Online]. Available: https://www.ibm.com/docs/en/zos/2.4.0?topic=overview-address-space-layout-randomization
[108] ‘SELinux Wiki’. Accessed: Aug. 24, 2024. [Online]. Available: https://selinuxproject.org/page/Main_Page
[109] ‘What is Mandatory Access Control? | NordLayer Learn’. Accessed: Aug. 27, 2024. [Online]. Available: https://nordlayer.com/learn/access-control/mandatory-access-control/
[110] L. Foster, ‘How to Strengthen the Linux Kernel Against Attacks’, Medium. Accessed: Aug. 27, 2024. [Online]. Available: https://medium.com/@lfoster49203/how-to-strengthen-the-linux-kernel-against-attacks-55b6595e938c
[111] ‘Hardware abstraction layer (HAL) overview’, Android Open Source Project. Accessed: Aug. 27, 2024. [Online]. Available: https://source.android.com/docs/core/architecture/hal
[112] ‘Sensors in Smartphones to Top 10 Billion Unit Shipments in 2020 - Counterpoint’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.counterpointresearch.com/insights/sensors-smartphones-top-10-billion-unit-shipments-2020/
[113] N. D. Wanjari and Shailaja. C. Patil, ‘Wearable devices’, in *2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT)*, Dec. 2016, pp. 287–290. doi: 10.1109/ICAECCT.2016.7942600.

[114] ‘Android Camera Hardware Architecture explained’, Minhaz’s Blog. Accessed: Aug. 27, 2024. [Online]. Available: https://blog.minhazav.dev/android-camera-hardware-explained/
[115] ‘What is a microphone?’, WhatIs. Accessed: Aug. 27, 2024. [Online]. Available: https://www.techtarget.com/whatis/definition/microphone
[116] ‘Position sensors | Sensors and location’, Android Developers. Accessed: Aug. 27, 2024. [Online]. Available: https://developer.android.com/develop/sensors-and-location/sensors/sensors_position
[117] E. Products, ‘Magnetometer basics for mobile phone applications’, Electronic Products. Accessed: Aug. 27, 2024. [Online]. Available: https://www.electronicproducts.com/magnetometer-basics-for-mobile-phone-applications/
[118] ‘GPS Working, Types, Examples’, Spiceworks Inc. Accessed: Aug. 27, 2024. [Online]. Available: https://www.spiceworks.com/tech/iot/articles/what-is-gps/
[119] F. Anthony, ‘Best Android Phones With LiDAR: Android’s Finest’, Aosmark. Accessed: Aug. 27, 2024. [Online]. Available: https://aosmark.com/best-android-phones-with-lidar/
[120] ‘Smartphone Accelerometers: Uses and How They Work’. Accessed: Aug. 27, 2024. [Online]. Available: https://www.fizziq.org/en/post/how-does-my-smartphone-s-accelerometer-work
[121] ‘Fingerprint HIDL’, Android Open Source Project. Accessed: Aug. 27, 2024. [Online]. Available: https://source.android.com/docs/security/features/authentication/fingerprint-hal
[122] ‘Touch devices’, Android Open Source Project. Accessed: Aug. 27, 2024. [Online]. Available: https://source.android.com/docs/core/interaction/input/touch-devices
[123] ‘Barometric Pressure Sensor Basics | Pressure Sensor’, Murata Manufacturing Co., Ltd. Accessed: Aug. 27, 2024. [Online]. Available: https://www.murata.com/en-eu/products/sensor/pressure/overview/basic
[124] Packt, ‘Step Detector and Step Counters Sensors’, Packt Hub. Accessed: Aug. 27, 2024. [Online]. Available: https://hub.packtpub.com/step-detector-and-step-counters-sensors/
[125] ‘Environment sensors | Sensors and location’, Android Developers. Accessed: Aug. 27, 2024. [Online]. Available: https://developer.android.com/develop/sensors-and-location/sensors/sensors_environment
[126] ‘Sensor types’, Android Open Source Project. Accessed: Aug. 27, 2024. [Online]. Available: https://source.android.com/docs/core/interaction/sensors/sensor-types
[127] ‘What is the Hall Sensor on a Mobile Phone? What are the purposes? - ZTE’. Accessed: Aug. 27, 2024. [Online]. Available: https://support.ztedevices.com/en-us/what-is-the-hall-sensor-on-a-mobile-phone-what-are/
[128] ‘Smart Geiger Counter Radiation Meter Android iPhone SMG - Specialist , 49,99 €’, Smart Geiger Counter Radiation Meter Android iPhone SMG - Specialist , 49,99 €. Accessed: Aug. 27, 2024. [Online]. Available: https://top-messtechnik.com/Smart-Geiger-Counter-SMG
[129] ‘Lux | Light Measurement, Photometry & Illumination | Britannica’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.britannica.com/science/lux
[130] ‘Ambient Light Sensor Market Is Predicted To Surpass USD’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.globenewswire.com/en/news-release/2023/03/22/2632005/0/en/Ambient-Light-Sensor-Market-Is-Predicted-To-Surpass-USD-15-7-Billion-at-a-CAGR-of-10-70-by-2030-Report-by-Market-Research-Future-MRFR.html
[131] ‘Consumer Electronics - Worldwide | Statista Market Forecast’, Statista. Accessed: Aug. 27, 2024. [Online]. Available: https://www.statista.com/outlook/cmo/consumer-electronics/worldwide
[132] ‘Automotive & Mobility - Ambient Light Sensing - ams-osram - ams’, ams-osram. Accessed: Aug. 24, 2024. [Online]. Available: https://ams-osram.com/applications/automotive-mobility/ambient-light-sensing
[133] ‘Six Commonly Used Sensors of LED Smart Lighting’, AGC Lighting. Accessed: Aug. 24, 2024. [Online]. Available: https://www.agcled.com/blog/six-commonly-used-sensors-of-led-smart-lighting.html
[134] D. Ben-Zeev, E. A. Scherer, R. Wang, H. Xie, and A. T. Campbell, ‘Next-generation psychiatric assessment: Using smartphone sensors to monitor behavior and mental health’, *Psychiatr. Rehabil. J.*, vol. 38, no. 3, pp. 218–226, 2015, doi: 10.1037/prj0000130.

[135] ‘Use of Sensors in Industrial Lighting Applications’, DigiKey. Accessed: Aug. 27, 2024. [Online]. Available: https://www.digikey.ch/en/articles/use-of-sensors-in-industrial-lighting-applications
[136] ‘Photoelectric effect | Definition, Examples, & Applications | Britannica’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.britannica.com/science/photoelectric-effect
[137] ‘Phototransistor : Construction, Circuit Diagram & Its Applications’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.elprocus.com/phototransistor-basics-circuit-diagram-advantages-applications/
[138] ‘From Photoresistors to Photodiodes - Types of Light Sensors and Their Applications - Quarktwin Electronic Parts’, Quarktwin Electronic. Accessed: Aug. 24, 2024. [Online]. Available: https://www.quarktwin.com/blogs/sensor/from-photoresistors-to-photodiodes-types-of-light-sensors-and-their-applications/101
[139] ‘What is a Photointerrupter? | Electronics Basics | ROHM’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.rohm.com/electronics-basics/photointerrupters/what-is-a-photointerrupter
[140] ‘Using CdS Photoresistor, Thermistors Theory’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.bristolwatch.com/ele/pd.html
[141] ‘Color Sensing Improves Look and Feel of Smart Products’.
[142] ‘Silicon Photomultiplier (SiPM) Structure, Characteristics, and Applications - Technical Articles’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.allaboutcircuits.com/technical-articles/silicon-photomultiplier-structure-characteristics-and-applications/
[143] R. MarketingUSA, ‘Ambient Light Sensor (ALS) Applications in Portable Electronics’.
[144] P. Bhat and K. Dutta, ‘A Survey on Various Threats and Current State of Security in Android Platform’, *ACM Comput Surv*, vol. 52, no. 1, p. 21:1-21:35, Feb. 2019, doi: 10.1145/3301285.
[145] ‘An Insight into Android Side-Channel Attacks | IEEE Conference Publication | IEEE Xplore’. Accessed: Aug. 24, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/8947838?denied=
[146] M. Nauman, S. Khan, and X. Zhang, ‘Apex: extending Android permission model and enforcement with user-defined runtime constraints’, in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, in ASIACCS ’10. New York, NY, USA: Association for Computing Machinery, Apr. 2010, pp. 328–332. doi: 10.1145/1755688.1755732.
[147] S. Bugiel, L. Davi, A. Dmitrienko, T. Fischer, and A.-R. Sadeghi, ‘XManDroid: A New Android Evolution to Mitigate Privilege Escalation Attacks’.
[148] Jun Han, E. Owusu, L. T. Nguyen, A. Perrig, and J. Zhang, ‘ACComplice: Location inference using accelerometers on smartphones’, in *2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012)*, Bangalore, India: IEEE, Jan. 2012, pp. 1–9. doi: 10.1109/COMSNETS.2012.6151305.
[149] ‘Motion sensors | Sensors and location’, Android Developers. Accessed: Aug. 24, 2024. [Online]. Available: https://developer.android.com/develop/sensors-and-location/sensors/sensors_motion
[150] S. Narain, T. D. Vo-Huu, K. Block, and G. Noubir, ‘Inferring User Routes and Locations Using Zero-Permission Mobile Sensors’, in *2016 IEEE Symposium on Security and Privacy (SP)*, San Jose, CA: IEEE, May 2016, pp. 397–413. doi: 10.1109/SP.2016.31.
[151] Z. Xu, K. Bai, and S. Zhu, ‘TapLogger: inferring user inputs on smartphone touchscreens using on-board motion sensors’, in *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*, in WISEC ’12. New York, NY, USA: Association for Computing Machinery, Apr. 2012, pp. 113–124. doi: 10.1145/2185448.2185465.
[152] A. Maiti, M. Jadliwala, J. He, and I. Bilogrevic, ‘(Smart)Watch Your Taps: Side-Channel Keystroke Inference Attacks using Smartwatches’, Sep. 2015, pp. 27–30. doi: 10.1145/2802083.2808397.
[153] Y. M. D. Boneh and G. Nakibly, ‘Gyrophone: Recognizing Speech From Gyroscope Signals’.

[154] ‘AccelWord | Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services’. Accessed: Aug. 24, 2024. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/2742647.2742658
[155] J. L. Fitch and A. Holbrook, ‘Modal Vocal Fundamental Frequency of Young Adults’, *Arch. Otolaryngol. - Head Neck Surg.*, vol. 92, no. 4, pp. 379–382, Oct. 1970, doi: 10.1001/archotol.1970.04310040067012.
[156] L. Bedogni, M. Felice, and L. Bononi, ‘By train or by car? Detecting the user’s motion type through Smartphone sensors data’, Nov. 2012, pp. 1–6. doi: 10.1109/WD.2012.6402818.
[157] W. Wu, S. Dasgupta, E. Ramirez, C. Peterson, and G. Norman, ‘Classification Accuracies of Physical Activities Using Smartphone Motion Sensors’, *J. Med. Internet Res.*, vol. 14, p. e130, Oct. 2012, doi: 10.2196/jmir.2208.
[158] Y. Kwon, K. Kang, and C. Bae, ‘Unsupervised learning for human activity recognition using smartphone sensors’, *Expert Syst. Appl.*, vol. 41, no. 14, pp. 6067–6074, Oct. 2014, doi: 10.1016/j.eswa.2014.04.037.
[159] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, ‘Fusion of Smartphone Motion Sensors for Physical Activity Recognition’, *Sensors*, vol. 14, no. 6, Art. no. 6, Jun. 2014, doi: 10.3390/s140610146.
[160] L. Zhang, J. Liu, H. Jiang, and Y. Guan, ‘SensTrack: Energy-Efficient Location Tracking With Smartphone Sensors’, *IEEE Sens. J.*, vol. 13, no. 10, pp. 3775–3784, Oct. 2013, doi: 10.1109/JSEN.2013.2274074.
[161] I. Griswold-Steiner, R. Matovu, and A. Serwadda, ‘Handwriting watcher: A mechanism for smartwatch-driven handwriting authentication’, in *2017 IEEE International Joint Conference on Biometrics (IJCB)*, Oct. 2017, pp. 216–224. doi: 10.1109/BTAS.2017.8272701.
[162] I. Griswold-Steiner, R. Matovu, and A. Serwadda, ‘Wearables-Driven Freeform Handwriting Authentication’, *IEEE Trans. Biom. Behav. Identity Sci.*, vol. 1, no. 3, pp. 152–164, Jul. 2019, doi: 10.1109/TBIOM.2019.2912401.
[163] F. Ciuffo and G. M. Weiss, ‘Smartwatch-based transcription biometrics’, in *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, Oct. 2017, pp. 145–149. doi: 10.1109/UEMCON.2017.8249014.
[164] A. Maiti, R. Heard, M. Sabra, and M. Jadliwala, ‘Towards Inferring Mechanical Lock Combinations using Wrist-Wearables as a Side-Channel’, in *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, in WiSec ’18. New York, NY, USA: Association for Computing Machinery, Jun. 2018, pp. 111–122. doi: 10.1145/3212480.3212498.
[165] A. Mylonas, V. Meletiadis, L. Mitrou, and D. Gritzalis, ‘Smartphone sensor data as digital evidence’, *Comput. Secur.*, vol. 38, pp. 51–75, Oct. 2013, doi: 10.1016/j.cose.2013.03.007.
[166] A. Das, N. Borisov, and M. Caesar, ‘Tracking Mobile Web Users Through Motion Sensors: Attacks and Defenses’, in *Proceedings 2016 Network and Distributed System Security Symposium*, San Diego, CA: Internet Society, 2016. doi: 10.14722/ndss.2016.23390.
[167] W. by, ‘Your Android’s accelerometer could be used to eavesdrop on your calls’, Sophos News. Accessed: Aug. 24, 2024. [Online]. Available: https://news.sophos.com/en-us/2019/07/23/spearphone-researchers-eavesdrop-on-phone-loudspeakers/
[168] S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi, ‘AccelPrint: Imperfections of Accelerometers Make Smartphones Trackable’, in *Proceedings 2014 Network and Distributed System Security Symposium*, San Diego, CA: Internet Society, 2014. doi: 10.14722/ndss.2014.23059.
[169] H. Bojinov, Y. Michalevsky, G. Nakibly, and D. Boneh, ‘Mobile Device Identification via Sensor Fingerprinting’, Aug. 06, 2014, *arXiv*: arXiv:1408.1416. Accessed: Aug. 24, 2024. [Online]. Available: http://arxiv.org/abs/1408.1416
[170] M. Azizyan, I. Constandache, and R. Roy Choudhury, ‘SurroundSense: mobile phone localization via ambience fingerprinting’, in *Proceedings of the 15th annual international conference on Mobile computing and networking*, in MobiCom ’09. New York, NY, USA: Association for Computing Machinery, Sep. 2009, pp. 261–272. doi: 10.1145/1614320.1614350.

[171] C. Shen, S. Pei, Z. Yang, and X. Guan, ‘Input extraction via motion-sensor behavior analysis on smartphones’, *Comput. Secur.*, vol. 53, pp. 143–155, Sep. 2015, doi: 10.1016/j.cose.2015.06.013.
[172] A. Maiti, M. Jadliwala, J. He, and I. Bilogrevic, ‘Side-Channel Inference Attacks on Mobile Keypads using Smartwatches’, Oct. 10, 2017, *arXiv*: arXiv:1710.03656. Accessed: Aug. 24, 2024. [Online]. Available: http://arxiv.org/abs/1710.03656
[173] H. Wang, T. T.-T. Lai, and R. Roy Choudhury, ‘MoLe: Motion Leaks through Smartwatch Sensors’, in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, Paris France: ACM, Sep. 2015, pp. 155–166. doi: 10.1145/2789168.2790121.
[174] X. Liu, Z. Zhou, W. Diao, Z. Li, and K. Zhang, ‘When Good Becomes Evil: Keystroke Inference with Smartwatch’, in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, in CCS ’15. New York, NY, USA: Association for Computing Machinery, Oct. 2015, pp. 1273–1285. doi: 10.1145/2810103.2813668.
[175] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury, ‘Tapprints: your finger taps have fingerprints’, in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, Low Wood Bay Lake District UK: ACM, Jun. 2012, pp. 323–336. doi: 10.1145/2307636.2307666.
[176] L. Cai and H. Chen, ‘TouchLogger: Inferring Keystrokes On Touch Screen From Smartphone Motion’.
[177] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, ‘ACCessory: password inference using accelerometers on smartphones’, in *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications - HotMobile ’12*, San Diego, California: ACM Press, 2012, p. 1. doi: 10.1145/2162081.2162095.
[178] M. Mehrnezhad and E. Toreini, ‘What Is This Sensor and Does This App Need Access to It?’, *Informatics*, vol. 6, no. 1, Art. no. 1, Mar. 2019, doi: 10.3390/informatics6010007.
[179] L. Simon and R. Anderson, ‘PIN skimmer: inferring PINs through the camera and microphone’, in *Proceedings of the Third ACM workshop on Security and privacy in smartphones & mobile devices*, in SPSM ’13. New York, NY, USA: Association for Computing Machinery, Nov. 2013, pp. 67–78. doi: 10.1145/2516760.2516770.
[180] T.-H. Do and M. Yoo, ‘An in-Depth Survey of Visible Light Communication Based Positioning Systems’, *Sensors*, vol. 16, no. 5, Art. no. 5, May 2016, doi: 10.3390/s16050678.
[181] M. Backes, T. Chen, M. Duermuth, H. P. A. Lensch, and M. Welk, ‘Tempest in a Teapot: Compromising Reflections Revisited’, in *2009 30th IEEE Symposium on Security and Privacy*, May 2009, pp. 315–327. doi: 10.1109/SP.2009.20.
[182] M. Backes, M. D, and D. Unruh, ‘Compromising Reflections-or-How to Read LCD Monitors around the Corner’, in *2008 IEEE Symposium on Security and Privacy (sp 2008)*, Oakland, CA, USA: IEEE, May 2008, pp. 158–169. doi: 10.1109/SP.2008.25.
[183] M. G. Kuhn, ‘Optical time-domain eavesdropping risks of CRT displays’, in *Proceedings 2002 IEEE Symposium on Security and Privacy*, Berkeley, CA, USA: IEEE Comput. Soc, 2002, pp. 3–18. doi: 10.1109/SECPRI.2002.1004358.
[184] L. Schwittmann, C. Boelmann, V. Matkovic, M. Wander, and T. Weis, ‘Identifying TV Channels & On-Demand Videos using Ambient Light Sensors’, *Pervasive Mob. Comput.*, vol. 38, pp. 363–380, Jul. 2017, doi: 10.1016/j.pmcj.2016.08.018.
[185] L. Schwittmann, V. Matkovic, M. Wander, and T. Weis, ‘Video recognition using ambient light sensors’, in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Mar. 2016, pp. 1–9. doi: 10.1109/PERCOM.2016.7456511.
[186] Y. Xu, J.-M. Frahm, and F. Monrose, ‘Watching the Watchers: Automatically Inferring TV Content From Outdoor Light Effusions’, in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, Scottsdale Arizona USA: ACM, Nov. 2014, pp. 418–428. doi: 10.1145/2660267.2660358.
[187] Y. Chen, T. Li, R. Zhang, Y. Zhang, and T. Hedgpeth, ‘EyeTell: Video-Assisted Touchscreen Keystroke Inference from Eye Movements’, in *2018 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA: IEEE, May 2018, pp. 144–160. doi: 10.1109/SP.2018.00010.
[188] J. Sun, X. Jin, Y. Chen, J. Zhang, R. Zhang, and Y. Zhang, ‘VISIBLE: Video-Assisted Keystroke Inference from Tablet Backside Motion’, in *Proceedings 2016 Network and*

Distributed System Security Symposium, San Diego, CA: Internet Society, 2016. doi: 10.14722/ndss.2016.23060.

[189] A. Holmes, S. Desai, and A. Nahapetian, 'LuxLeak: capturing computing activity using smart device ambient light sensors', in *Proceedings of the 2nd Workshop on Experiences in the Design and Implementation of Smart Objects*, in SmartObjects '16. New York, NY, USA: Association for Computing Machinery, Oct. 2016, pp. 47–52. doi: 10.1145/2980147.2980150.

[190] A. Maiti and M. Jadliwala, 'Light Ears: Information Leakage via Smart Lights', *Proc ACM Interact Mob Wearable Ubiquitous Technol*, vol. 3, no. 3, p. 98:1-98:27, Sep. 2019, doi: 10.1145/3351256.

[191] E. Ronen and A. Shamir, 'Extended Functionality Attacks on IoT Devices: The Case of Smart Lights', in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, Mar. 2016, pp. 3–12. doi: 10.1109/EuroSP.2016.13.

[192] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, 'Luxapose: indoor positioning with mobile phones and visible light', in *Proceedings of the 20th annual international conference on Mobile computing and networking*, Maui Hawaii USA: ACM, Sep. 2014, pp. 447–458. doi: 10.1145/2639108.2639109.

[193] Z. Wang, Z. Yang, Q. Huang, L. Yang, and Q. Zhang, 'ALS-P: Light Weight Visible Light Positioning via Ambient Light Sensor', in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, Apr. 2019, pp. 1306–1314. doi: 10.1109/INFOCOM.2019.8737575.

[194] B. Xie, G. Tan, Y. Liu, M. Lu, K. Chen, and T. He, 'LIPS: A Light Intensity Based Positioning System For Indoor Environments', Mar. 07, 2014, *arXiv*: arXiv:1403.2331. doi: 10.48550/arXiv.1403.2331.

[195] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, 'Epsilon: A Visible Light Based Positioning System', presented at the 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14), 2014, pp. 331–343. Accessed: Aug. 24, 2024. [Online]. Available: https://www.usenix.org/conference/nsdi14/technical-sessions/presentation/li

[196] N. Rajagopal, P. Lazik, and A. Rowe, 'Visual light landmarks for mobile devices', in *Proceedings of the 13th international symposium on Information processing in sensor networks*, in IPSN '14. Berlin, Germany: IEEE Press, Apr. 2014, pp. 249–260.

[197] P. Lou, H. Zhang, X. Zhang, M. Yao, and Z. Xu, 'Fundamental analysis for indoor visible light positioning system', in *2012 1st IEEE International Conference on Communications in China Workshops (ICCC)*, Aug. 2012, pp. 59–63. doi: 10.1109/ICCCW.2012.6316475.

[198] M. Biagi, A. M. Vegni, and T. D. C. Little, 'LAT indoor MIMO-VLC — Localize, access and transmit', in *2012 International Workshop on Optical Wireless Communications (IWOW)*, Pisa, Italy: IEEE, Oct. 2012, pp. 1–3. doi: 10.1109/IWOW.2012.6349698.

[199] C. M. Avendaño-Lopez, R. Castro-Sanchez, D. L. Almanza-Ojeda, J. G. Avina-Cervantes, M. A. Gomez-Martinez, and M. A. Ibarra-Manzano, 'Scalable Visible Light Indoor Positioning System Using RSS', *Mathematics*, vol. 10, no. 10, Art. no. 10, Jan. 2022, doi: 10.3390/math10101738.

[200] F. Alam, M. T. Chew, T. Wenge, and G. S. Gupta, 'An Accurate Visible Light Positioning System Using Regenerated Fingerprint Database Based on Calibrated Propagation Model', *IEEE Trans. Instrum. Meas.*, vol. 68, no. 8, pp. 2714–2723, Aug. 2019, doi: 10.1109/TIM.2018.2870263.

[201] Y. Hu *et al.*, 'Lightitude: Indoor Positioning Using Uneven Light Intensity Distribution', *Proc ACM Interact Mob Wearable Ubiquitous Technol*, vol. 2, no. 2, p. 67:1-67:25, Jul. 2018, doi: 10.1145/3214270.

[202] Z. Yang, Z. Wang, J. Zhang, C. Huang, and Q. Zhang, 'Wearables Can Afford: Light-weight Indoor Positioning with Visible Light', in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, in MobiSys '15. New York, NY, USA: Association for Computing Machinery, May 2015, pp. 317–330. doi: 10.1145/2742647.2742648.

[203] Z. Zhao, J. Wang, X. Zhao, C. Peng, Q. Guo, and B. Wu, 'NaviLight: Indoor localization and navigation under arbitrary lights', in *IEEE INFOCOM 2017 - IEEE Conference on Computer*

Communications, Atlanta, GA, USA: IEEE, May 2017, pp. 1–9. doi: 10.1109/INFOCOM.2017.8057184.
[204] Q. Xu, R. Zheng, and S. Hranilovic, 'IDyLL: indoor localization using inertial and light sensors on smartphones', in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, in UbiComp '15. New York, NY, USA: Association for Computing Machinery, Sep. 2015, pp. 307–318. doi: 10.1145/2750858.2807540.
[205] X. Zhu, W. Qu, T. Qiu, L. Zhao, M. Atiquzzaman, and D. O. Wu, 'Indoor Intelligent Fingerprint-Based Localization: Principles, Approaches and Challenges', *IEEE Commun. Surv. Tutor.*, vol. 22, no. 4, pp. 2634–2657, 2020, doi: 10.1109/COMST.2020.3014304.
[206] J. Armstrong, Y. A. Sekercioglu, and A. Neild, 'Visible light positioning: a roadmap for international standardization', *IEEE Commun. Mag.*, vol. 51, no. 12, pp. 68–73, Dec. 2013, doi: 10.1109/MCOM.2013.6685759.
[207] Y. Wang *et al.*, 'Spectral-Loc: Indoor Localization Using Light Spectral Information', *Proc ACM Interact Mob Wearable Ubiquitous Technol*, vol. 7, no. 1, p. 37:1-37:26, Mar. 2023, doi: 10.1145/3580878.
[208] Y. Liu, G. W. Wornell, W. T. Freeman, and F. Durand, 'Imaging privacy threats from an ambient light sensor', *Sci. Adv.*, vol. 10, no. 2, p. eadj3608, Jan. 2024, doi: 10.1126/sciadv.adj3608.
[209] 'Samsung Galaxy S10+ - Full phone specifications'. Accessed: Aug. 27, 2024. [Online]. Available: https://www.gsmarena.com/samsung_galaxy_s10+-9535.php
[210] 'Android Version Market Share Worldwide', StatCounter Global Stats. Accessed: Aug. 24, 2024. [Online]. Available: https://gs.statcounter.com/os-version-market-share/android
[211] 'Samsung Galaxy S10+ popularity & fame | YouGov'. Accessed: Aug. 24, 2024. [Online]. Available: https://today.yougov.com/topics/technology/explore/model_of_phone/Samsung_Galaxy_S10
[212] 'Light Sensors | ams'. Accessed: Aug. 24, 2024. [Online]. Available: https://ams.com/en/light-sensors
[213] 'Architecture overview', Android Open Source Project. Accessed: Aug. 24, 2024. [Online]. Available: https://source.android.com/docs/core/architecture
[214] 'Sensor Logger — Kelvin Choi'. Accessed: Aug. 24, 2024. [Online]. Available: https://www.tszheichoi.com/sensorlogger
[215] 'Android Debug Bridge (adb) | Android Studio | Android Developers'. Accessed: Aug. 24, 2024. [Online]. Available: https://developer.android.com/tools/adb
[216] 'Machine Learning and Biometrics System - Javatpoint'. Accessed: Aug. 24, 2024. [Online]. Available: https://www.javatpoint.com/machine-learning-and-biometric-systems
[217] 'What is Supervised Learning?', Google Cloud. Accessed: Aug. 24, 2024. [Online]. Available: https://cloud.google.com/discover/what-is-supervised-learning
[218] 'https://uk.mathworks.com/campaigns/offers/next/choosing-the-best-machine-learning-classification-model-and-avoiding-overfitting.html'. Accessed: Aug. 24, 2024. [Online]. Available: https://uk.mathworks.com/campaigns/offers/next/choosing-the-best-machine-learning-classification-model-and-avoiding-overfitting.html
[219] P. Barjatiya, 'Unleashing the Power of Random Forest: Why it Outperforms Decision Trees and Expert Rules', Medium. Accessed: Aug. 24, 2024. [Online]. Available: https://pratikbarjatya.medium.com/unleashing-the-power-of-random-forest-why-it-outperforms-decision-trees-and-expert-rules-472a9bea1b8a
[220] A. Nagpal and G. Gabrani, 'Python for Data Analytics, Scientific and Technical Applications', in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, Feb. 2019, pp. 140–145. doi: 10.1109/AICAI.2019.8701341.
[221] 'Why Do Data Analysts Use Python?', UCD Professional Academy. Accessed: Aug. 24, 2024. [Online]. Available: https://www.ucd.ie/professionalacademy/resources/why-do-data-analysts-use-python/
[222] 'pandas - Python Data Analysis Library'. Accessed: Aug. 24, 2024. [Online]. Available: https://pandas.pydata.org/
[223] 'NumPy -'. Accessed: Aug. 24, 2024. [Online]. Available: https://numpy.org/
[224] 'SciPy -'. Accessed: Aug. 24, 2024. [Online]. Available: https://scipy.org/

[225] ‘scikit-learn: machine learning in Python — scikit-learn 1.5.1 documentation’. Accessed: Aug. 24, 2024. [Online]. Available: https://scikit-learn.org/stable/
[226] ‘Weka 3 - Data Mining with Open Source Machine Learning Software in Java’. Accessed: Aug. 24, 2024. [Online]. Available: https://ml.cms.waikato.ac.nz/weka/
[227] ‘Visual Studio: IDE and Code Editor for Software Developers and Teams’. Accessed: Aug. 24, 2024. [Online]. Available: https://visualstudio.microsoft.com/
[228] ‘Project Jupyter’. Accessed: Aug. 24, 2024. [Online]. Available: https://jupyter.org
[229] ‘colab.google’. Accessed: Aug. 24, 2024. [Online]. Available: https://colab.google/
[230] ‘PyCharm: the Python IDE for data science and web development’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.jetbrains.com/pycharm/
[231] ‘Open for Innovation | KNIME’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.knime.com/
[232] ‘Cloud Computing Services | Microsoft Azure’. Accessed: Aug. 24, 2024. [Online]. Available: https://azure.microsoft.com/en-gb
[233] *weka: A Python wrapper for the Weka data mining library.* Python. Accessed: Aug. 24, 2024. [OS Independent]. Available: https://gitlab.com/chrisspen/weka
[234] ‘(PDF) WEKA: The Waikato Environment for Knowledge Analysis’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.researchgate.net/publication/2703103_WEKA_The_Waikato_Environment_for_Knowledge_Analysis
[235] ‘Arff stable - Weka Wiki’. Accessed: Aug. 24, 2024. [Online]. Available: https://waikato.github.io/weka-wiki/formats_and_processing/arff_stable/
[236] ‘CSV, Comma Separated Values (RFC 4180)’. Accessed: Aug. 26, 2024. [Online]. Available: https://www.loc.gov/preservation/digital/formats/fdd/fdd000323.shtml
[237] ‘Bayes’ Theorem’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.mathsisfun.com/data/bayes-theorem.html
[238] ‘Evaluation’. Accessed: Aug. 24, 2024. [Online]. Available: https://weka.sourceforge.io/doc.dev/weka/classifiers/evaluation/Evaluation.html
[239] ‘ISO 27001 – Annex A.9: Access Control | ISMS.online’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.isms.online/iso-27001/annex-a-9-access-control/
[240] ‘SensorManager | Android Developers’. Accessed: Aug. 24, 2024. [Online]. Available: https://developer.android.com/reference/android/hardware/SensorManager
[241] ‘What Are Naïve Bayes Classifiers? | IBM’. Accessed: Aug. 24, 2024. [Online]. Available: https://www.ibm.com/topics/naive-bayes
[242] N. Khanna, ‘J48 Classification (C4.5 Algorithm) in a Nutshell’, Medium. Accessed: Aug. 24, 2024. [Online]. Available: https://medium.com/@nilimakhanna1/j48-classification-c4-5-algorithm-in-a-nutshell-24c50d20658e
[243] ‘Logistic Regression in Machine Learning’, GeeksforGeeks. Accessed: Aug. 24, 2024. [Online]. Available: https://www.geeksforgeeks.org/understanding-logistic-regression/
[244] ‘Hyperparameter Tuning — Determined AI Documentation’. Accessed: Aug. 24, 2024. [Online]. Available: https://docs.determined.ai/latest/model-dev-guide/hyperparameter/_index.html
[245] ‘Notepad++’. Accessed: Aug. 26, 2024. [Online]. Available: https://notepad-plus-plus.org/
[246] C. O. Back, S. Debois, and T. Slaats, ‘Entropy as a Measure of Log Variability’, *J. Data Semant.*, vol. 8, no. 2, pp. 129–156, Jun. 2019, doi: 10.1007/s13740-019-00105-3.
[247] S. L. Faulkner and S. P. Trotter, ‘Data Saturation’, in *The International Encyclopedia of Communication Research Methods*, John Wiley & Sons, Ltd, 2017, pp. 1–2. doi: 10.1002/9781118901731.iecrm0060.

Appendix

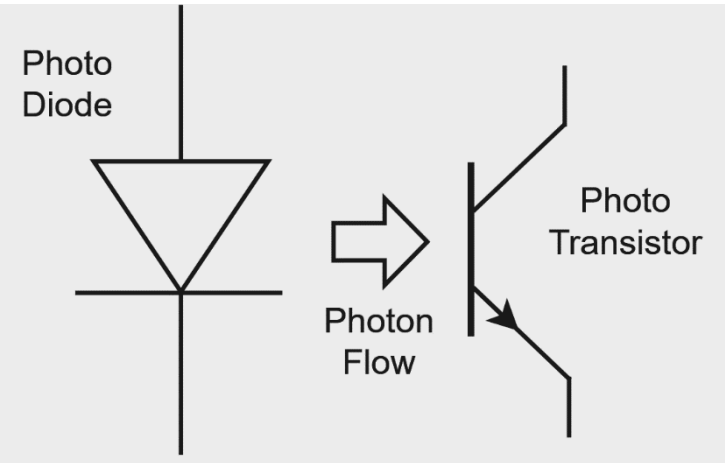

Figure 48: Depiction of a Photo Interrupter

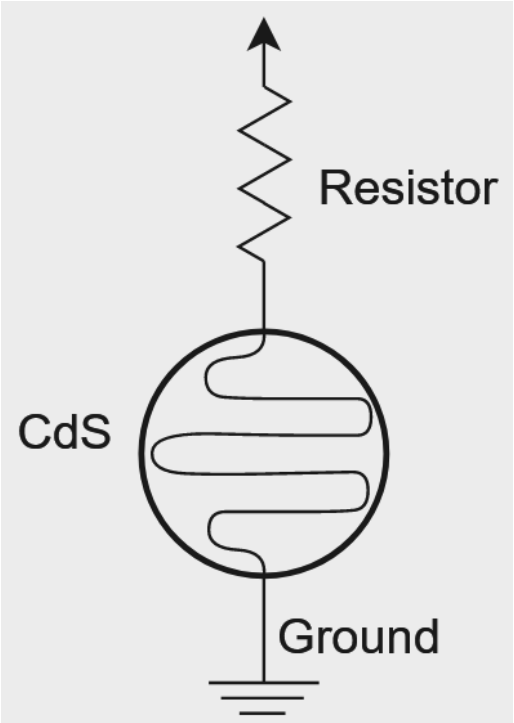

Figure 49: depiction of a CdS cell

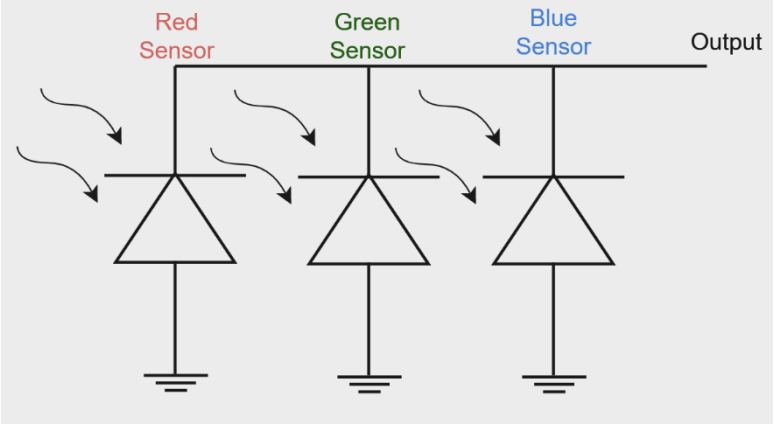

Figure 50: depiction of RGB colour sensors

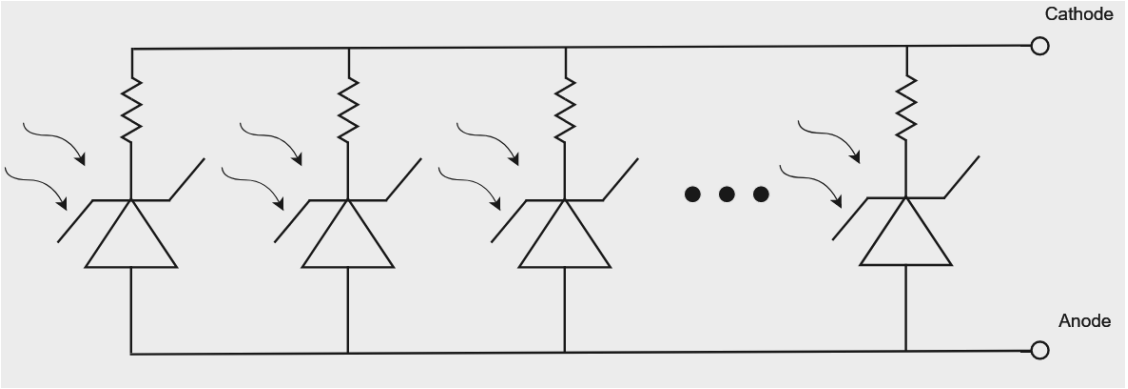

Figure 51: depiction of silicon photo multipliers

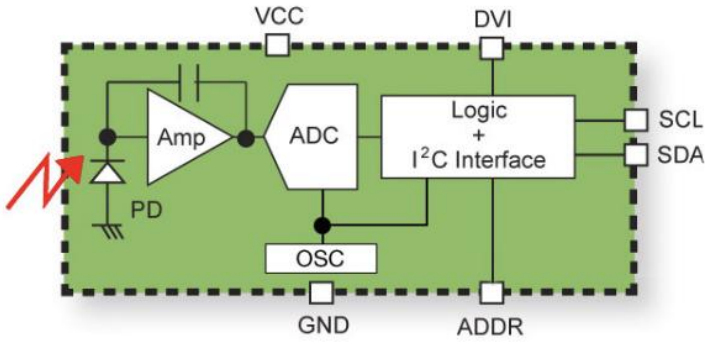

Figure 52: Digital Photo ICs [121]

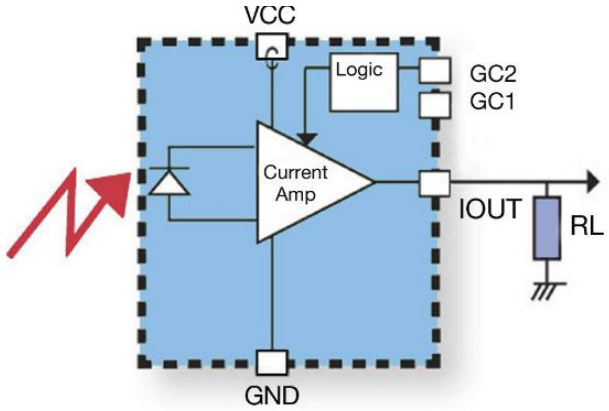

Figure 53: layout of an analogue photo ICs [121]

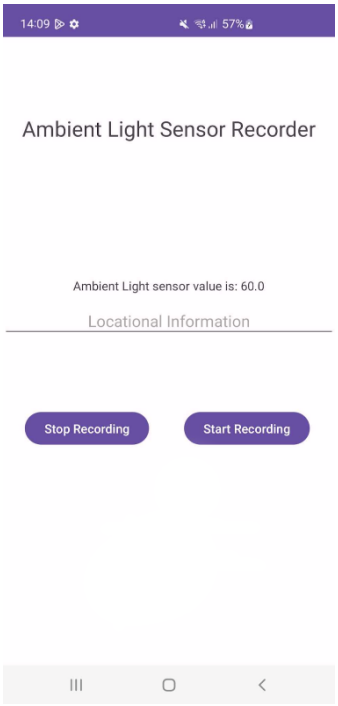

Figure 54: ALS Recorder Application UI

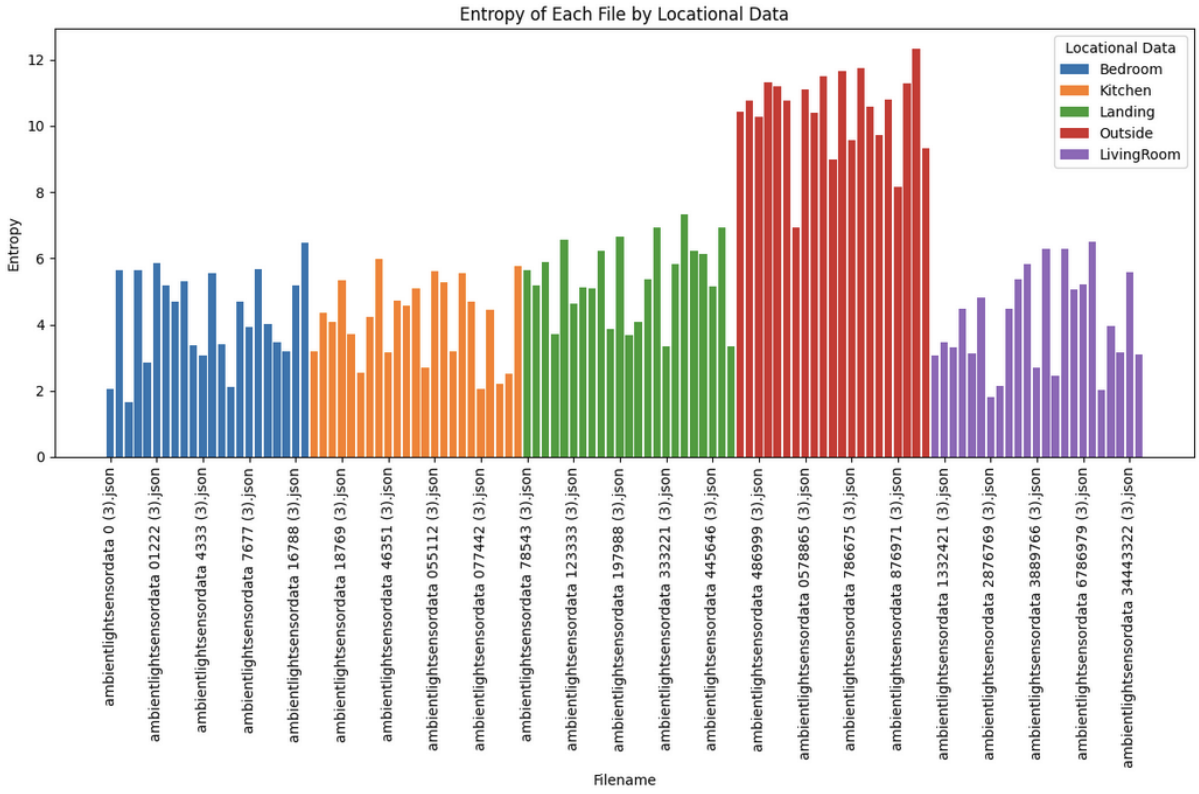

Figure 55: Entropy of Each File by Locational Data