Improving Real-World Accuracy and Data Efficiency in Indoor Light Sensing for Multifunctional Applications

by

Tushar Kanti Routh

A dissertation document submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
in

Department of Electrical and Computer Engineering
University of Virginia
April, 2025

Doctoral Committee:

Bradford Campbell, (Advisor), Associate Professor, Computer Science & Electrical and Computer Engineering
Lu Feng, Associate Professor, Computer Science & Engineering Systems and Environment
Cong Shen, Associate Professor: Electrical & Computer Engineering
Tariq Iqbal, (Chair), Assistant Professor, Systems Engineering & Computer Science
Jonathan L. Goodall, Professor, Civil and Environmental Engineering

APPROVAL SHEET

This

Dissertation

is submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Author: Tushar Kanti Routh

Advisor: Dr. Bradford Campbell

Advisor:

Committee Member: Dr. Tariq Iqbal

Committee Member: Dr. Jonathan L. Goodall

Committee Member: Dr. Cong Shen

Committee Member: Dr. Lu Feng

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Jennifer L. West, School of Engineering and Applied Science

April 2025

ABSTRACT

In recent years, there has been an expansion of sensor applications, which includes sensing indoor environments through smart indoor light sensors. These sensors can help explore human daylong indoor light exposure, energy efficient lighting adjustments, reorganize the interior based on guidelines, or personify indoor illumination. However, data collected from these sensors to date exhibit poor accuracy in realistic scenarios, like identification of indoor light sources under unknown sources, under the presence of multiple-sources, or switching from one light to another. Furthermore, the presence of surrounding noise or fluctuations in sensor readings due to irrelevant events can adversely affect classification accuracy and lead to unnecessary resource consumption. Unfortunately, current sensing and classification techniques are data and power inefficient for daylong sensing. Moreover, understanding the capabilities of indoor light sensors and analyzing potential security issues is essential for building secure and privacy-preserving indoor environments.

In this dissertation, we aim to explore several key approaches to address these challenges and enhance real-world classification accuracy. First, we propose generating synthetic and filtered datasets to replicate real-world scenarios and provide a more comprehensive set of examples beyond controlled environments. This helps the classifier become more adept at handling diverse situations encountered in real-life settings. To enable data-efficient robust daylong source identification, we propose multiple approaches. We first introduce the dimension reduction technique to eliminate unnecessary overhead information. Secondly, we develop an intelligent on-device algorithm capable of detecting light source transitions and facilitating time-specific exposure identification. Third, we present `SENTREC`, a platform designed to identify the most robust and accurate segment within a long sequence of sensed values, at the same time, capable of differentiating between targeted and non-targeted events. Finally, we introduce `ScreenSense`, a framework that utilizes basic color information from indoor light sources for identifying users' activities on digital screens. This framework provides a low-power and enhanced privacy solution for monitoring daylong screen activity, as well as educating smart building professionals regarding potential security risks associated with improper installation of these sensors in smart indoor environments.

ACKNOWLEDGEMENTS

I owe this life milestone to the unwavering support and encouragement of my teachers, colleagues, family, and friends. I would first like to express my heartfelt gratitude to my advisor, Dr. Bradford Campbell. His encouragement, guidance, and mentorship have been crucial to my success. He is, without a doubt, the best advisor I have had the privilege of working with throughout my journey and I strongly believe that he exemplifies the qualities of an ideal PhD advisor. Dr. Campbell will always remain my academic role model, and I aspire to inspire others in the same way he has inspired me.

I am deeply grateful to my committee members, Dr. Tariq Iqbal, Dr. Cong Shen, Dr. Lu Feng, and Dr. Jonathan Goodall, who have provided invaluable insights and expertise from a range of disciplines, greatly enhancing my dissertation and shaping my ideas. From the LMI team, I would like to express my heartfelt thanks to Robert Wagner and Michael Brennan for their insightful advice.

I would not have been able to complete my PhD without the unwavering support, sacrifice and enthusiasm of my family: my mother Shikha, my late uncle Mrinal, my aunt Rekha, and my brother Shishir. The contributions of Md. Monsurul Huda, Mahjabin Maksud Mishi, Nahid, and Shuvro are beyond measure, and it is impossible to fully repay their support. I am deeply grateful to Samiul, Mishu, Tibtum, Nazmul, and Barunda for their invaluable support and contributions during my initial years in the United States. On this occasion, I would like to express my sincere gratitude to my mentor, Dr. Rob Maher, whose contributions to this journey have been unparalleled.

My PhD journey has been a challenging and rewarding roller coaster that has spanned over eight years. My experience in Charlottesville would not have been nearly as enriching if it weren't for the presence of my amazing friends and seniors Abu Mondol, Anindya Prodhan, Ifat Emi, Jashim Uddin, Amzad, Nahid and Azim. Finally, I am eternally grateful to my wife, Anonya Chowdhury, whose unwavering love, support, and sacrifices have made this journey possible.

I extend my sincere gratitude to my fellow colleagues and friends, Rabbi, Cynthia, Nabeel, Victor, Jiechao, Viswajith, Fateme, Wenpeng, Marshall, and Alex, whose insightful discussions and stimulating academic environment were invaluable throughout my doctoral journey.

I am deeply grateful to the faculty and staff of the College of Engineering at the University of Virginia, especially Beth Eastwood-Beatty. Their dedication to fostering a supportive research environment has been instrumental to my academic achievements.

All of your presence in my life has been an immense source of strength and I wouldn't have made it this far without you. Thank you for being a part of my journey.

To my parents and my wife, for their continuous support over the years.

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Through general perception and evaluation, it has been established that *vision* is the most important sense of the human. The principal element of human vision is *light*, produced by both natural and artificial light sources, helps us conceptualize our surroundings while entering the human visionary system [1]. A general-purpose light sensor can sense lighting parameters in its surroundings and provide light-sensitive data that can be processed for various applications, such as promoting wellness, improving energy efficiency, optimizing setups, ensuring compliance with regulatory guidelines, and so on.

A light sensor detects and measures the intensity of light by converting optical signals into electrical signals using photodetectors such as photodiodes or phototransistors. It enables real-time monitoring and analysis of indoor lighting conditions for various applications, including automation, health monitoring, and energy efficiency [2]. In recent years, indoor light sensors have shown their value beyond just managing indoor lighting. They have the potential to drive innovative, human-focused solutions. Researchers have interlinked the influence of different types of screen activities with human behavior analysis [3]. For such analysis, a light sensor can come into play to unfold the platform-specific information of a screen user without affecting too much screen privacy. A basic solar cell, capable of generating power according to the availability of light, can serve as a light sensor and indicate various indoor activities through particular fluctuations in generated voltage. These techniques pave the way for low-power solutions to address the aforementioned tasks while expanding the potential to utilize existing or already deployed sensing devices for diverse applications, significantly contributing to the advancement of Green IoT.

However, using these sensors to classify events has some real-world limitations. Although classifiers can exhibit high accuracy in controlled environments, their performance declines when encountering

unfamiliar real-world variations due to improper training, restricting their widespread adoption. Moreover, in long-term sensing applications, the absence of smart sensing or data reduction techniques results in excessive data generation, causing unnecessary energy consumption and frequent power replacements.

This dissertation explores the challenges faced by various indoor light sensors for different applications, focusing on declination of real-world classification accuracy and generation of overhead information for continuous, daylong operation. It presents solutions to improve real-world classification performance and support broader adoption by retraining the classifiers with an augmented dataset that simulates real-world variations. It also presents data reduction methods and intelligent algorithms that can be implemented on these sensors to identify and collect event specific information needed for classification while filtering out irrelevant data, enabling long-term sensing application in indoor environments.

1.1 Motivation for Indoor Lighting Sensing

Daylong indoor mobility exposes inhabitants to natural and various types of artificial lights, which operate at specific wavelengths and generate illumination of contrasting features. Recent studies have explored how daylong light exposure affects human physiology by examining factors such as heart rate, cortisol levels, core body temperature, fatigue, and sleep patterns [4]. Daylong indoor mobility exposes inhabitants to natural and various types of artificial lights, which operate at specific wavelengths and generate illumination of distinct features, to which our sensory systems react in different ways. Exposure to inappropriate lighting conditions have been associated with disruptive circadian rhythms and behavioral problems in people with dementia [5]. Sleep disorders, affecting 50 to 70 million adults and one-third of the senior population in the U.S.A., are often associated with irregular melatonin production caused by prolonged or inconsistent light exposure [6], [7]. Not only lighting parameters, but also lighting type, plays a critical role here. Blue-enriched light sources such as modern LEDs can suppress and delay the natural functioning of the biological clock [8]. For that, health professionals recommend avoiding blue-enriched sources after sunset hours to obtain quality sleep [9]. Furthermore, research shows that well-designed lighting can improve the well-being of seniors, Alzheimer's patients, and residents who spend a certain number of hours at indoors [10]–[12]. Smart lighting control is therefore crucial for maintaining occupant comfort and

can also reduce operating costs in buildings [13].

The swift advancement and adoption of smart technologies and sensing systems have opened up numerous possibilities for technological progress across various facets of life [14]. As a result, more than 20 billion sensing devices are predicted to be connected currently in the whole world. It has also been estimated that the global energy consumption of sensing edge devices will approach 46TWh by 2025 [15]. Light sensors have become a key component in the sensor market, experiencing significant growth driven by their integration into smart homes, wearables, and various other devices. This growth is further fueled by advancements in miniaturization and the rising demand for smart city applications. The total market is expected to grow around $7.63 Bn. by 2030 in the USA at a CAGR of 12% [16]. In addition to meeting the visual need of inhabitants, maintaining certain lighting environments is essential to highlight consumer products/artworks, controlling agro-environments, detection of biochemicals, heavy metals, environmental nutrients and so on [17]–[19].

For green sensor based applications, the goal should be: (a) utilizing as much low energy as possible through smart operation design and (b) leveraging a single device/collected data from a particular observation for multiple applications. In indoor environments, we should explore opportunities to leverage indoor light sensors for additional applications other than light source detection. For example, fluctuations in the values detected by a light sensor can offer valuable insights into indoor activities, potentially removing the need for a dedicated activity sensor for security or monitoring purposes [20].

While enabling diverse applications in shared domestic and commercial settings, the widespread adoption of smart indoor light sensors largely relies on factors such as consistent performance and data/energy efficiency, meaning the desired performance can be attained while utilizing a substantially reduced amount of sensed data for post-processing. Unfortunately, existing light sensing techniques suffer from significant limitations, including unnecessary energy consumption, overhead data generation, and poor performance in real-world scenarios. This necessitates a critical need for improved solutions.

1.2 Application Variability and Associated Requirements

In real-world scenarios, the number of sensors needed at a time, sensor type, and operation duration can vary according to various indoor environments and applications. For example, Occupational

Safety and Health Administration (OSHA) has standards for the choice of light types in places like classrooms with students having Autism Spectrum Disorders (ASDs) or at building stairs for better wayfinding and spatial perception for people with severe vision impairments. Monitoring the surrounding lighting scenario can be completed with a single-sensor, operated only for a limited duration to establish the appropriate lighting environment at the point of interest.

Inhabitants may want to calculate the number of daily hours in blue-enriched sources (*like LED)* for correlating nighttime sleep quality with blue light hours or for documenting the natural light disclosure period to meet the minimum daily recommended vitamin D generation. A single light sensor is also sufficient there, but the sensing duration spans from dawn to dusk. A wearable device may be suitable for particular individuals, but for patients/senior citizens who have limited movements and are already carrying wearable medical devices, a fixed point light sensor is a better alternative.

Indoor light sensors for continuous monitoring of illumination is particularly essential for places like hospitals/auditoriums/multipurpose arenas, where rearrangement takes place too often. Not only multiple light sensors may be required here, but they may need to be operated in a wireless manner, powered with batteries and need to be placed at specific corners that can be difficult to visit too often.

Light sensors can be used for applications beyond their intended purpose, allowing them to gather color-sensitive data from sources other than artificial room lights or natural sunlight. Indoor light sensors positioned near a computer screen can capture variations corresponding to the activity displayed on the screen. These distinctive patterns can then be analyzed to detect screen activity without exposing extraneous browsing details which can severely impact users privacy. As a result, indoor light sensors can be an energy and data efficient alternative for privacy preserving screen activity analysis. Collected color sensitive screen data from these devices can later be utilized for the physiological and psychological study related to on-screen behavior, associating social website hours on learning/social interactions/sleep duration, prediction of human personalities and so on [21]. On the other hand, smart building experts should have a better idea regarding the capabilities of indoor light sensors. Proper placement is crucial to prevent these devices from inadvertently exposing privacy-sensitive information. For this application, low-cost light sensors are essential, as each monitor will require its own sensing device. Additionally, the sensors must offer the flexibility to be deployed either as a wearable or a fixed-point device near the workstation.

For gathering color sensitive information for the aforementioned purposes, choice of appropriate sensor is crucial, as certain types have limitations for certain applications. For example, mini spectrometers were used by experts for identifying daylong source exposure. For classification, daylong spectrum data were utilized, where the power supply and the data storage medium needed to get replaced in every couple of hours. In addition, particular spectral band information for classification has some limitations, as same source type, (like *LEDs* for example) or even the same source (like sun throughout the day and at diverse weather conditions/indoor scenarios) can exhibit variation in spectral information. as a result, spectrum based information is unsuitable for the above-mentioned applications.

Over the past two decades, solar energy harvesting technology has advanced rapidly, transitioning from supporting space programs to becoming an integral part of daily life [22]. Solar cells, also known as photovoltaic cells, function as a particular type of light sensor, generating an output voltage proportional to the amount of light incident on their exposed surfaces. Recent studies have demonstrated that when solar cells can also function effectively as an independent sensor based on voltage variations, like temperature sensors [23] or occupancy detectors [24]. For that, it is critical to explore efficient ways to extract and use information from these sensors for our intended purpose.

To integrate all these requirements of indoor light sensing for different applications, light sensors need to meet the deployment criteria of being inexpensive, deployment flexibility, and importantly, power efficient for daylong operation, especially under power budget. One potential solution to improve power efficiency is by utilizing only the essential components for sensing and offloading the energy-intensive processing tasks to other platforms. All those necessities drive the choice of IoT-based light sensing, which meets the key criteria outlined, as explored in the following section.

1.3 IoT-Enabled Sensors as a Solution

In recent years, Internet of Things (IoT) based light sensors have been increasingly deployed in residential and commercial buildings. IoT-based sensors are physical devices equipped with sensing capabilities and connected to the Internet of Things (IoT) ecosystem [25]. Their versatility in smart systems, cost-effectiveness, and potential applications in security and personalized lighting make them crucial for both residential and commercial settings on a mass scale.

Through wireless or wired connectivity, IoT-based sensors communicate with cloud-based systems

or other devices, enabling real-time monitoring, analysis, and automation across diverse applications, including smart homes, industrial automation, healthcare, and environmental monitoring [26]. These include *ambient sensors* for adjusting brightness, *proximity sensors* for detecting the presence or absence of objects, and *LiDAR sensors* for 3D mapping and ranging, among others. After addition of sensing devices *(for example, motion sensors, temperature sensor etc.)*, these smart devices are capable of advertising the sensed on formation over a Wireless Sensor Network (WSN), that is reinforced by low-cost and lower power devices, such as Wi-Fi, Bluetooth, Zigbee, Near Frequency Communication, etc. [27].

IoT-based light sensing offers several advantages for long-term, daylong operability in indoor environments. These sensors can optimize energy efficiency, enhance comfort, and continuously adjust lighting conditions to align with user preferences. For example, an inhabitant can simply wear this sensor as a smart watch throughout the day. Based on that, he/she can calculate the number of daily hours in blue-enriched sources (*like LED)* for correlating nighttime sleep quality with blue light hours or for documenting natural light disclosure period to meet the minimum daily recommended vitamin D generation [28], [29]. For long-term analysis and power/memory-intensive classification of various types of sources throughout the day, these sensors can also transmit the sensed data to a remote platform, offloading the resource-intensive tasks of data storage and classification.

Based on the capability of sensed parameters, IoT-based light sensors come in various types, including simple lux-based sensors, infrared (IR) sensors, color temperature sensors, multi-spectrum light sensors, and so on. Additionally, a solar cell can be integrated into the IoT platform, functioning as a light sensor for indoor event detection, such as occupant identification or exit/entry event monitoring [24]. Now, *which particular IoT based sensor is best suited for purposes like identification of daylong sources or screen activity detection?* For that, the sensor should include only the basic and essential components for low-power operation, while being cost-effective, easy to deploy, and capable of generating memory-friendly information that is sufficient for long-term operation and classification. For mass adoption, the focus should be on leveraging widely available sensor types or existing sensor frameworks rather than developing new sensor prototypes.

For seamless integration into smart devices, **IoT-based RGB color sensors** have gained significant popularity in recent times [30]. Equipped with basic color-specific filter elements, these sensors are compact and versatile, making them well-suited for deployment as either wearable or fixed-point

devices. They are designed to operate seamlessly without interfering with the most commonly used sensing platforms at indoor environments and can function in both wired and wireless configurations, ensuring flexibility in their application. Instead of collecting in-details spectral information, most general IoT based color sensors can record only basic RGB values and intensity information. Their cost-effectiveness and extremely low energy consumption during operation have made them widely adopted in smart homes for applications such as energy-efficient lighting design, illumination adjustment, and lighting condition monitoring.

When placed near any type of indoor light source, these devices are capable of recording different RGB patterns over a period, which is sufficient to classify sources of different types. In indoor environments, these RGB patterns can also be detected from nearby computer screens. As different activity on screen and for the same activity, different people can record signature RGB patterns from screen, an *IoT enabled RGB based indoor light sensor*, which is primarily utilized for indoor lighting, can be an effective tool for passive detection of screen user and screen activity. *That is why for daylong light source exposure analysis and passive screen activity detection, we choose* ***IoT based RGB sensors*** *for analysis.*

As discussed previously, a photovoltaic cell can function as an occupant identifier. When positioned near an entrance, the movement of an individual can disrupt the ambient light, causing a detectable perturbation. This disturbance generates a specific voltage variation, which can serve as a unique signature pattern for identifying that individual. As these cells contain only sensing element with no processing circuitry, *for our investigation, we have seamlessly integrated a solar cell with an IoT platform.* This platform can store the recorded voltage data and transmit it to a remote system for occupant detection over the course of a day.

1.4 Current Challenges and Limitations of Indoor Light Sensing

Although IoT based light sensors meet challenges in terms of cost, low energy operation, and flexibility of installation, several critical challenges must still be addressed before their effective deployment in real-world applications.

Although these sensors generate only RGB and intensity information that is both memory-efficient and adequate for the aforementioned applications, continuous and non-adaptive sensing over extended periods can lead to an accumulation of memory-intensive overhead data. This results in

unnecessary power consumption and inefficient use of processing resources. As a result, the power supplies need to get replaced frequently, which is not always practical. Moreover, current techniques are inadequate for applications that require long-term operation within a limited energy budget. For instance, if an individual remains under the same light source for several hours, repeatedly detecting the same source for daylong exposure assessment is highly inefficient. Hence, it is essential to develop techniques that can reduce information overhead. Since the sensor's energy consumption depends on the number of data transmissions required for classification, reducing the data needed for classification minimizes transmissions and conserves energy. This can be achieved by (a) avoiding repetitive information generated during static/irrelevant period, (b) selectively capturing events of interests and transmitting only the information required for classifying that event and (c) compacting necessary information so that it can fit into a lower number of advertisement packets.

When identifying an occupant using voltage ripple data from a photovoltaic sensor, the target events are brief, making continuous data processing unnecessary and energy-intensive. Similar voltage fluctuations can arise from surrounding noise sources or non-targeted events, potentially leading to unnecessary data processing and information overhead. For instance, a basic event such as dragging a chair near a solar cell can alter the light availability, creating a disturbance that resembles occupant movement. However, this is an event that the user may not be interested in classifying. A data efficient way is to filter out such events for post-processing. To date, no intelligent approach has been developed that can accurately identify a targeted event from a lengthy sequence of sensed values, ensuring resilience against surrounding noise, and at the same time, minimize data overhead.

When discussing classification, classifiers trained on a limited set of examples struggle to recognize variations that arise from real-world scenarios. This happens for several reasons, such as unseen light sources, transient patterns due to random switching, and interference from other nearby indoor light sources. These variations differ significantly from those encountered in limited and controlled lighting environments, resulting in poor identification and decreased classification accuracy. Therefore, it is essential to first generate examples that reflect these real-world variations based on the available dataset and use them to re-train the classifiers. This will allow the classifiers to recognize the unfamiliar realistic variations and maintain consistent classification accuracy.

This PhD dissertation addresses the aforementioned unresolved challenges that hinder the effective implementation of indoor light sensors in real-world environments. While these challenges have primarily been identified in the context of indoor light sensing, they are also relevant to other types

of sensor data exhibiting similar characteristics.

1.5 Thesis Statement

Sensing indoor lighting environments can not only educate occupants regarding physical and psychological well-being but also identify inhabitants' identity and their acts on digital screens. For multifunctional operation, sensing and classification techniques to-date utilize redundant information. In addition, limited training set and incidents like switching of indoor light sources or the presence of multiple sources declines the classification accuracy in real-world. By generating synthetic/filtered datasets to mimic real-world variants of examples from controlled environments and introducing dimension reduction/specific sample selection for classification, the elimination of overhead information is possible, and real-world classification accuracy can be improved.

1.6 Real-World challenges related to Classification with Indoor Light Sensors

In the next section, we discuss a few key challenges, apart from those mentioned earlier for the real-world implementation of indoor light sensors for multiple purposes. The complete list of challenges addressed in particular projects has been listed in Figure 6.1 To provide a comprehensive understanding of these challenges and the corresponding solutions, we present a detailed table summarizing the key aspects of the problems and their proposed resolutions in 1.1. Finally, we evaluate the effectiveness of the proposed techniques by comparing the scenarios before and after their implementation in 1.2.

1.6.1 Challenges regarding Sensing

Deployment: Apart from recording daylong source exposure, indoor light sensors are also utilized to continuous monitor whether a certain lighting criteria has been maintained at the place of interest or not, especially. To achieve this, in addition to being utilized as wearable devices, they are also deployed as fixed-point installations. Even with the best performing classifier, choice of deployment parameters like the distance from source and no of observations considered have profound impact on classifiers' performance, which has also not been analyzed for the optimal performance.

1.6.2 Challenges regarding Classification

Identifying single/multi source environments: Indoors can have a simultaneous presence of multi-type light sources. For example, places like photographic studios or art galleries utilize multi-type sources to bring out a particular luminous environment or continuous adjustments between natural and artificial sources are required for indoor plantation. To bring out a particular luminous environment, it is important to know what types of sources are present at the place of interest. Till to date, classifiers are only designed to identify a single source even in such multi-source environments.

Classification in a sensor-dense environment: Modern indoors are equipped with multi-type sensors. In such an environment, when an user wants to install a wireless light sensor and tries to offload information to a distant computer, there can be packet loss due to network congestion and communication bottleneck. Advertising fewer informative packets there is advantageous but can be traded-off with classification performance. How to condense color sensitive information and transfer them using fewer on-air transmissions and what are resource requirements for such method adaptation on-device, is yet to get analyzed.

1.6.3 Challenges regarding On-Screen Activity Detection

In this section, we discuss challenges while implementing indoor light sensor as a passive device for detecting screen activities. Since certain challenges overlap with those encountered in light source identification, they are not included here.

Identifying Irregular Patterns: For identifying on-screen platform, the operator can randomly switch from one platform to another within the time-frame of interest or can vary the playback speed. The user may also vary generate other variations, like changing the playback speed or playing a video in reverse. All those incidences generate RGB patterns that are unknown to the classifier. As of now, how to manually create these real-world variations from ideal scenarios and make the classifier familiar to them, has not been investigated.

Choosing Sampling frequency/Gain Settings: To operate sensors that are extremely resource-bound or in a scenario where energy supply to the sensor can experience fluctuations, collecting data at a high sampling rate only during the change in lighting scenario may not be enough, user may want to further minimize the required resources by collecting selective samples

for classification. This approach can not only shrinks the amount of information that stockpiles over a long period, but also elongate the operation duration of a sensor under limited energy availability. However, scaling down information can result into signals missing minute details of RGB variations essential for classification and which in turn, can degrade the classifying performance. When adjusting the amplification settings, higher levels allow signals to be captured from greater distances. However, this comes at the cost of increased power consumption, which reduces the sensor's operational lifespan under restricted supply. From engineering point of view, these are important to figure out before real-world deployment.

1.6.4 Challenge regarding Occupant detection with PV cells

Determining the most robust and confident segment In real-world scenarios, variations in the voltage of photovoltaic (PV) cells can arise from various factors, such as natural fluctuations in indoor light sources or interference from noise generated by other sensors. These disturbances can lead to fluctuations in the sensed values, which impact occupant identification performance. Therefore, it is essential to identify a segment that accurately represents the correct occupant class with high confidence, and does not get affected by such disturbances. Developing a universal approach to identify such segments based on both robustness and confidence presents a significant challenge, as it is highly dependent on the nature of the classifier and the type of sensed data. Additionally, the question of how to develop an approach that can detect these segments from streaming data has not yet been addressed in the existing literature.

1.7 Major Contributions

In this work, we begin by training our classifier using data collected from controlled environments. Once the classifier is trained, we help it adapt to real-world variations in two ways:

- We innovate and apply specialized filters to the controlled environment data. These filters are designed to simulate patterns that occur in real-world situations, such as changes caused by switching of light sources or random movements.
- We analyze the current data distribution and generate synthetic examples that represent scenarios the classifier has not yet encountered. This allows the classifier to recognize new

patterns and examples, making them suitable for universal real-world deployment.

After implementing those augmented dataset in the training set, the real world classification accuracy improved significantly in all the realistic testbeds.

Most indoor light sensors today are IoT-based and send data to a remote platform, offloading energy-intensive classification tasks to conserve energy at the sensor node. Excess energy is consumed when the sensor node advertises overhead data generating from sensed values irrelevant to the events/with limited variations, which is extremely important when sensor node needs to operate under energy budget. In this study, we present several data efficiency techniques that, while common in other domains, are entirely novel in this context.

- For example, while dimensionality reduction is commonly employed for information condensation, we reveal that existing variation in RGB data from indoor light sensors renders certain **non-linear dimension reduction technique** is effective in reorganizing meaningful information using only few dimensions.
- In addition, we observe that while a certain number of observations are necessary for classification, not all contribute equally to determining the correct class. Here, we reintroduce the **Permutation Feature Importance** technique can be introduced to rank samples based on their individual impact.
- We also introduce a **second-order differential method** to accurately differentiate an actual light switching events versus a non-switching natural variation of sunlight during sunrise and sunset.
- **Adversarial Attack Mechanisms**—commonly used to evaluate the robustness of machine learning models—are leveraged in this work to serve a novel purpose. We re-establish this concept for achieving data efficiency by identifying the most robust and informative segments within long recordings for event classification.
- We introduce a memory-efficient on-device filtration method that leverages a **KNN Segment Selector** to intelligently filter out events that are irrelevant or unintended for classification. This approach minimizes unnecessary data processing by focusing only on the most contextually significant segments from targeted events, thereby optimizing both computational resources and classification accuracy.

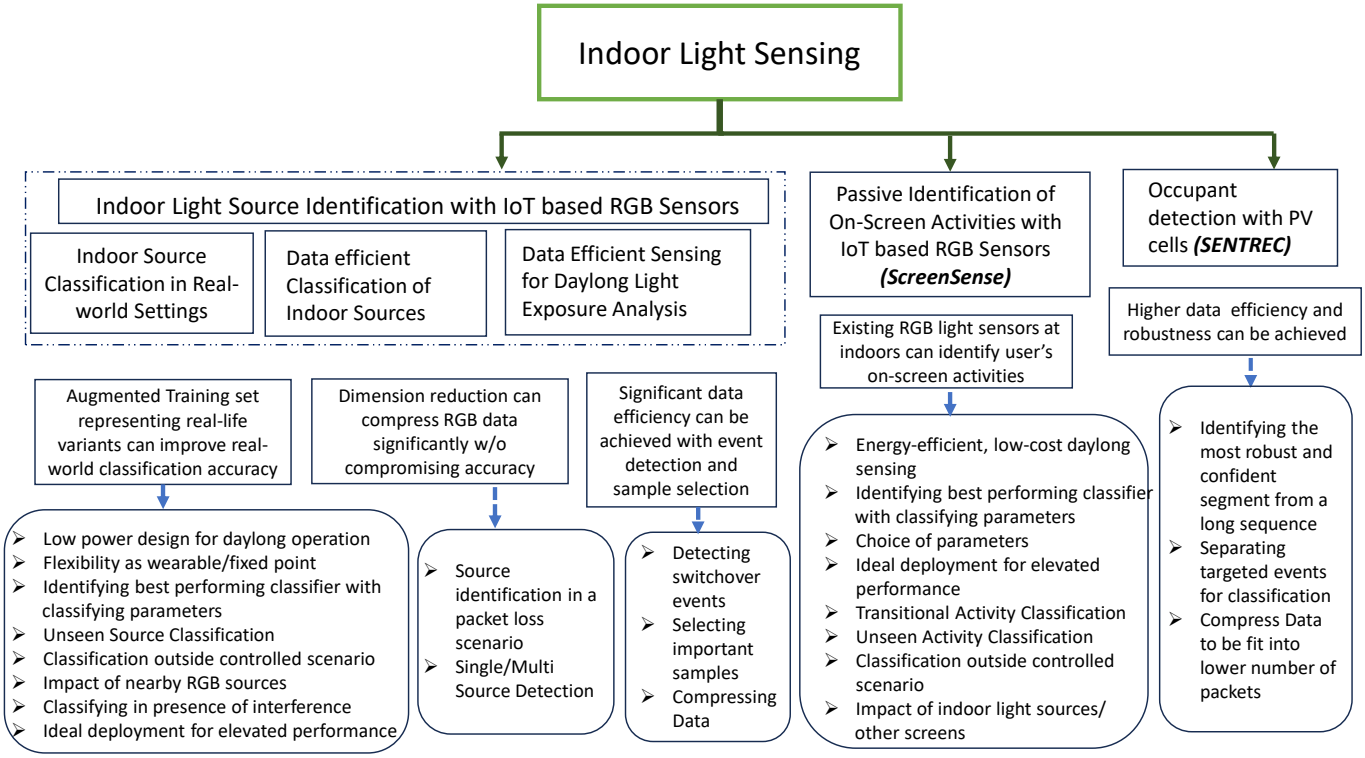


Figure 1.1: Conceptual integration of multiple projects addressed in this dissertation for analyzing RGB information based Indoor Light Source Classification/On-screen Activity Detection and Photo-voltage variation based Occupant Identification. Addressed key takeaways along with challenges associated with each project have also been demonstrated.

After investigation, we discover that after adopting these approaches, necessary information needed to advertise from the sensor node to the distant platform have been reduced greatly without compromising classification accuracy. As this reduces the number of advertisements, this will greatly help conserve energy and allow the sensors to operate for longer periods within energy restrictions.

Figure 6.1 illustrates the conceptual integration of our key contributions alongside the challenges addressed. These challenges are categorized into three main groups. The challenges related to indoor light source identification using RGB-based sensors have been tackled through three distinct sub-projects, each focusing on a specific set of real-world issues.

Although the proposed solutions are presented through the lens of indoor light sensors, they can be extended to broader sensor systems that face similar challenges. Along with outlining various tasks, limitations, challenges, and proposed solutions addressed in this work, table 1.1 also highlights applications of these solutions and their potential relevance to broader sensing systems. Table 1.2 presents a comparative analysis of classification accuracy and data efficiency before and after implementing our approaches.

Tasks	**Addressed Limitations**	**Challenges**	**Lacking**	**Proposed Solution**	**Other Applications**
Source Classification	Poor Classification Performance in Real-World Scenarios due to unknown patterns/unseen examples	Familiarize Classifier with transient patterns Making Classifier enable to detect unseen patterns from the same class	Inadequate Training set (lacks similar examples)	Augmented training through Synthetic Dataset (TimeGAN/ on-off dataset)	Random switching of source under investigation *(e.g.,monitoring household appliance usage)* Systems Requiring Universal Classification Performance *(e.g., Global Air Quality Monitoring)*
Source Classification	System performance varies due to deployment variations, features/classifier selection	Familiarize Classifier with deployment variations Understanding the underlying pattern of the sensor data for classification	Systematic study of deployment variations Studying data pattern first and then investigating different types of classifiers (ML, NN, Time Series)	Record performance fluctuations due to deployment variations and if needed, choose the best setting Selection of classifier based on data patterns *(like selecting KNN as the classifier where data exhibits locality)*	System performance can vary due to deployment factors *(e.g. indoor air quality monitoring)* Any sensing system where sensor data exhibits non-linearity and locality in feature space
Minimizing Data for Classification	Advertising is energy-expensive. Essential data can be reduced through Dimension Reduction method where data exhibits less variation in a time-frame	Selection of the appropriate method based on the linearity/non-linearity nature of sensor data	Studying various linear/non-linear technique for sensor data	Transform, Advertise and Reconstruct Data at the receiving end	Instances where high dimensional data with lower variations needs to be processed *(e.g., Flood Prediction through RGB camera)*
Data Efficient Daylong Sensing	Sensing the same surrounding environment/features repeatedly. In result, generating or dealing with overhead information	Isolating switch-over events from long recordings Pinpointing samples that are important for classification in a timeframe	A smart algorithm that can identify actual light switch-over events Implementing technique to sort samples in a timeframe based on their necessity for classification	A sensed parameter-based algorithm designed for seamless deployment on edge devices, capable of effectively identifying and distinguishing switch-over events Applied a feature importance-based approach to identify key samples within the timeframe for classification	Where systems need to differentiate between short-term variations caused by temporary disruptions vs long term trending *(e.g. Air Quality Monitoring with Gas Sensors)* Sensor systems that exhibits dissimilar patterns in particular samples (e.g. Analyzing ECG signals)

Tasks	**Addressed Limitations**	**Challenges**	**Lacking**	**Proposed Solution**	**Other Applications**
On Screen Activity Detection in presence of other RGB sources	Indoor Light Sources pollute the RGB signal from screen	Familiarize classifiers with screen activity patterns under light	Addressing the influence of nearby other sources that may generate similar sensor signals	Generate patterns under influence and retrain the classifier with those examples	Systems where multiple sources generate similar acoustic signals *(like in urban noise monitoring from different vehicles, sounds like loud machinery can produce signals with overlapping frequency ranges and signal patterns.*
Robust Segment Selection	Identifying the most robust and confident segment for a targeted event classification from a lengthy time series of sensor data.	Finding a universal robust and confident segment selection approach Developing a technique that can filter targeted events, is deployable on streaming data at edge devices	Focusing only on confidence, not robustness Lack of study regarding Filtration of targeted vs non-targeted events	Adversarial attack-based approach for robust segments/ Utilize surrogate models Developing a resource-efficient on-device classifier to distinguish targeted events	Sensor data where targeted events are categorized by certain type of bursts *(Physical Activity Detection through CSI, smart watch etc.)* Developing a resource-efficient on-device classifier to distinguish targeted events vs Non-targeted events *(Fall events and certain non-fall events (e.g., sitting down forcefully) produce similar acceleration and angular velocity patterns)*

Table 1.1: Overview of Major Addressed Limitations Across Multiple Projects related to Indoor Light Sensing: Current Challenges, Proposed Solutions, and Potential Applications in Other Sensor Systems

Title	Topic	Before Analysis	After Our Work
Source Identification	Classification Accuracy for Identification of Light Sources in Realistic Scenarios	65% (Outside Training Set) [31] 50% (switch on-off events) [32]	upto **90.25%** in unfamiliar & switch on/off settings testbed
Data Compression	RGB Data Compression for Source Classification with Dimension Reduction	300 bytes (25 RGB samples)	(2 principal components: 8 bytes) **97.33% Data Efficiency**
Daylong Source Exposure Detection	Compressing Daylong Data with Switching Detection and Lossless Compression	12000 bytes (300 Raw RGB samples)	247 bytes after detection and compression **(97.94% Data compressed)**
ScreenSense	Screen Activity Classification Accuracy	79.3% with Random forest 70.1% with Naive Bias	Testbed Accuracy upto **91.25%** [33]
SENTREC	Compression through Event Identification and Lossless Compression	1200 bytes (uncompressed data for 300 samples)	23 bytes of classification data **98.08% Data Efficiency**

Table 1.2: A comparative analysis of Major achievements regarding Classification accuracy and Data efficiency

Chapter 2

INDOOR LIGHT SENSING WITH BLE BASED RGB COLOR SENSING BOARD *(*LPCSB) AND PHOTOVOLTAIC CELL

As discussed earlier, the primary requirements for collecting daylong light information data are that the light sensor must consume minimal energy while being adaptable for use as either a wearable device or a fixed-point installation. For that, we utilize a Bluetooth Low Power (BLE) enabled color sensing board for acquiring light exposure information and on-screen color sensitive information for extended period and demonstrate how recording only RGB information can be fruitful identifying major source exposure at various times and different screen activities played at screen. This smart device exploits very low memory, suitable for indoor deployment and flexible to be shaped into wearable format if required. Based on sensed information, it can calculate on-the-spot lighting parameters like Lux Intensity (LI) and Correlated Color Temperature (CCT), as well as provide data to distinguish major source in background off-board. Although there are multiple color sensors in the market, most of them are not cost effective, are large dimensional, energy inefficient and require to relay information to central hub for further analysis mostly through wired connections. Our goal was to develop small scale, low-cost, mobile, lightweight task specific sensor, that is unobtrusive to already installed systems in that surroundings and easy to deploy as smart room sensor or as wearable systems in future. LPCSB advertises BLE packets containing RGB, clear value (related to intensity), color temperature and lux information of a light source (calculated from RGB values), which enables user to place the board in inaccessible/unreachable areas, connect with BLE receivers and then deliver sensed values as instructed. Moreover, the system consumes extremely low power, as a result the power source does not need to get replaced often which lowers down the maintenance hazards. In addition, information can be captured from a distance and analysed in any platform of users choice (for example, smart watch or remote servers). For

classification, we use this board only as an advertiser to advertise a BLE data packet containing ID, raw data (clear, red, green, and blue) split into two bytes per color, color temperature and lux of the measured light calculated from raw rgb values and the number of the latest packet being advertised.

LPCSB is a printed circuit board (PCB) that interfaces *TCS3475* sensor and is regulated with *nRF51822* micro controller. It is qualified to communicate over 2.4GHz Bluetooth Low Energy (BLE), flexible enough to operate in two way (transceiver mode) or one way (advertising) mode, as needed. System has a dimension of roughly $24mm \times 39.5mm$, suited to get fit and comfort as wearable devices. For low power consumption and simplification, *nRF51822* micro controller components were limited to only clock circuits, 3.3 V regulatory circuitry and power supply connector in the final design. *Micro Reach Xtend (FR05-S1-N-0-110) Chip Antenna* was assembled to establish communication and fit in PCB, plus USB connector for supplying power. Fully assembled LPCSB can be seen in Figure 7.4.

The power regulation section consisted of a micro-USB B-type connector, a green LED indicator circuit, MAX887EZK33+T Low-Dropout 300mA 3.3V Linear Regulator, and several bypass capacitors meant to help stabilize the input / output voltage and current in case of supply fluctuations. With the help of *BAL-NRF01D3 transformer balun* for impedance matching and ”LightBlue” phone app for monitoring, we have tested BLE radio transmitter inside *nRF51822*. Using the ”nrf5x-base” and ”Adafruit-TCS34725” GitHub repositories as design references, we have instructed the *TCS34725* through *nRF51822* to measure the ambient light and send the resulting values. Red-filtered, green-filtered, blue-filtered, and clear (unfiltered) diodes data of *TCS34725* sensor is stored as a 16-bit value, split between two registers. We have further calculated the color temperature of the light in degrees Kelvin and the lux in lumens per square meter, using formula provided by Adafruit. Figure 7.4 (d) represents energy intake per cycle of LPCSB, where sensor reading is followed by a BLE advertisement event. If we set parameters to classify source within a minute, avg current drawn is per sampling is around $0.22\mathrm{mA}$ and the system can operate upto 45 days with conventional 3.3V Lithium batteries without replacement.

To convert indoor light into electrical energy using the photovoltaic effect, we selected the **IXYS SMLD121H04L monocrystalline solar cell**. This high-efficiency, long-life silicon cell is specifically designed for applications requiring high reliability, extended service life, and minimal maintenance. It is well-suited for a variety of indoor and outdoor applications, including solar energy research, off-grid solar power systems, portable or mobile solar power systems, photovoltaic

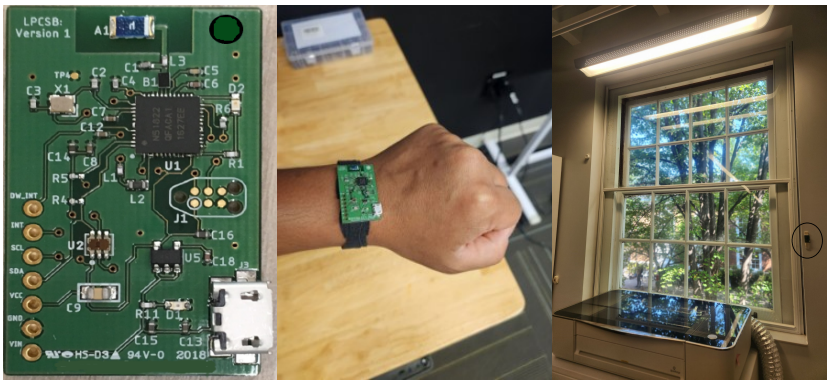
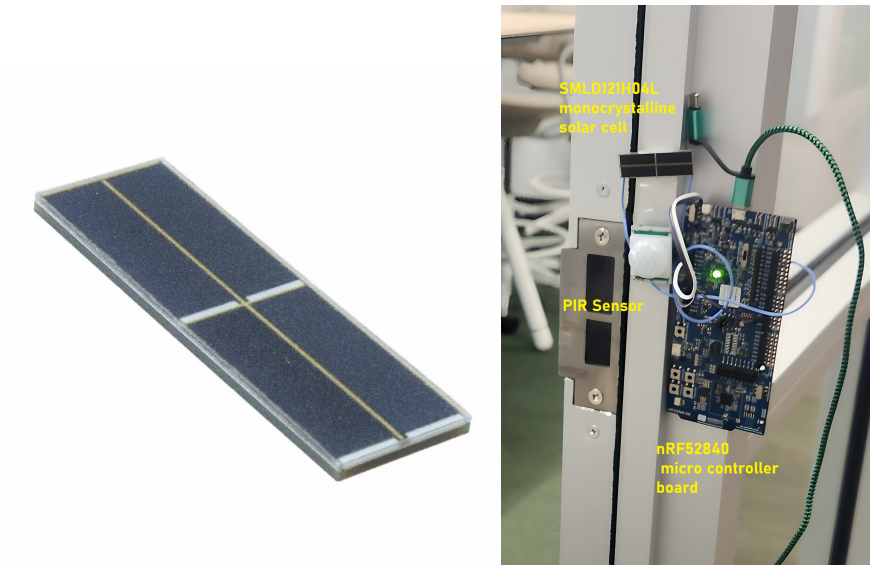


Figure 2.1: Fully assembled *BLE* enabled customised Low Power Color Sensing Board (a), Board Dimension is suitable to deploy as a handheld (b) or fixed point device (encircled) (c). Color Sensing followed with an advertisement event (d)

lighting systems, solar-powered road signs, security systems, water pumping systems, and traffic lights. The cell is engineered to optimize efficiency (with a reported accuracy of 22%) in a wide range of climatic conditions. Its open-circuit voltage is 2.52V, and it provides a short-circuit current of 50 mA. The dimensions of the solar cell are 43 x 14 mm [34].

To sample continuous voltage values, we use the same *nRF51822* micro controller board. At runtime, the MCU samples the solar cell at a 50Hz rate using one of the internal ADC channels. To capture a full event, the duration for collecting samples was set to 6 seconds. A Panasonic AMN41121 PIR sensor is utilized as a trigger generator to detect movement within a 5-meter range and a 50° horizontal field of detection at the doorway.



(a) **(b)**

Figure 2.2: SMLD121H04L monocrystalline solar cell (a) Deploying cell with MCU at the doorside(b)

Once the setup is complete, the solar cell surface is positioned perpendicular to the floor. During standard operation, the light intensity in indoor environments gradually varies throughout the day until the light source is switched off. However, the surrounding light intensity experiences a sudden change when someone passes nearby, which is reflected in the solar cell's output voltage. The solar cell along with the setup is shown in the Figure 2.2 .

Chapter 3

SYSTEMATIC STUDY OF DEPLOYMENT PARAMETERS FOR IDENTIFYING PRIME SOURCE AT INDOORS

This work focuses on identifying four types of indoor light sources with LPCSB, including the three most commonly used artificial bulb types: **Incandescent lamps, Compact Fluorescent Lights (CFLs), and Light Emitting Diodes (LEDs)—along with sunlight.** For sensing, LPCSB can be deployed flexibly to identify the primary light source, such as being worn as a wristband for personal daylong exposure analysis or installed at a fixed location to monitor specific light conditions in particular indoor spaces, like nursing rooms or auditoriums. Once deployed, LPCSB can capture RGB and brightness data from their surroundings. The collected information is then transmitted to a remote computer, which processes it using a pre-trained optimal classifier to generate and report the output.

In a controlled scenario, classification task is done with a fixed number of samples, with stand alone sources and the setups remained non interrupted throughout data collection. However, in real world, identifying environments can deviate from the ideal scenario in multiple ways. Modern day lighting architectures are not isolated, rather have become dynamic and personalized through blending sources of multiple types and specific features. In addition, modern indoor environments are equipped with smart systems that automate lighting by turning lights on or off based on the presence of an occupant or setup to seamlessly switch in between natural/artificial sources for setting up particular lighting scenario/achieve energy efficiency. These real-world incidents produce distinct RGB patterns compared to controlled scenarios, making classification significantly more challenging.

Determining the optimal classifier and the necessary number of samples is essential to achieve the

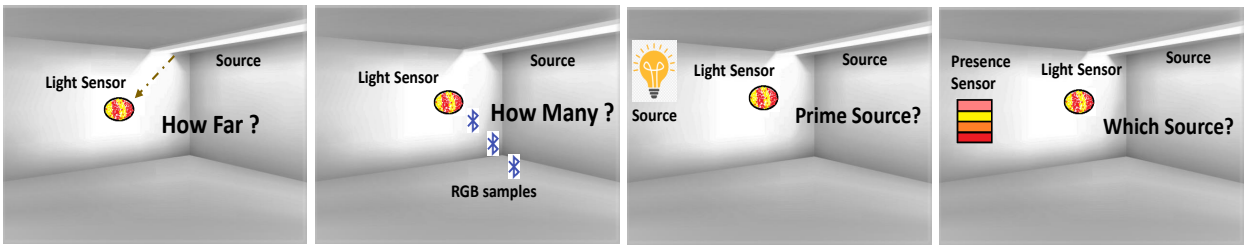


Figure 3.1: An illustration of deployment parameters studied here. From left to right a) Sensor placement, b) No of samples for classification, c) Detecting prime source in multi-source environment, d) Identifying light type at smart environments

best classification performance after data collection. In addition, when the user is interested in identifying the light source type of a particular location, he/she may want to place the light sensor at the particular position where the classifier yields the best performance. Even when the user is carrying a wearable light sensor, he/she is positioning himself/herself at different distances, which can impact the overall performance. Unfortunately, the overall performance of which classifier (along with the parameters) is the best (including both ideal and non-ideal scenarios) and how real-world uncontrolled variants can impact the indoor light source classifying performances have never been studied before.

To address these issues, we placed the sensor in indoor environments and collected color-sensitive data over a specific period, where varying light sources generate time-series data that create distinct patterns for classification. To address the real-world deployment challenges as mentioned, we investigate the ideal distance from light source to the sensors and the number of samples needed for optimal performance, along with their dimensions for optimal classifying performance. Our investigation also includes the prime source detection ranging from simple conditions, such as a single light source in a dark room, to complex environments with multiple light sources, smart switching, and noisy surroundings involving other RGB sources. To find the best performing classifier, we study multiple Machine Learning and Neural Network based classifying methods and made comparative analysis of accuracy of those classifiers in ideal/non-ideal backgrounds like multi-source/noisy/smart environments (Figure 3.1).

3.0.1 Why intensity is inadequate for understanding indoor lighting?

To demonstrate why only knowledge regarding light intensity is inadequate and how indoor source type can impact visual experience, we place LPCSB under two different types of light sources. Figure 3.2 shows how natural sunlight near the window and another spot very next to it, illuminated

with LED bulbs, result in a completely different visual experience with similar lux values.

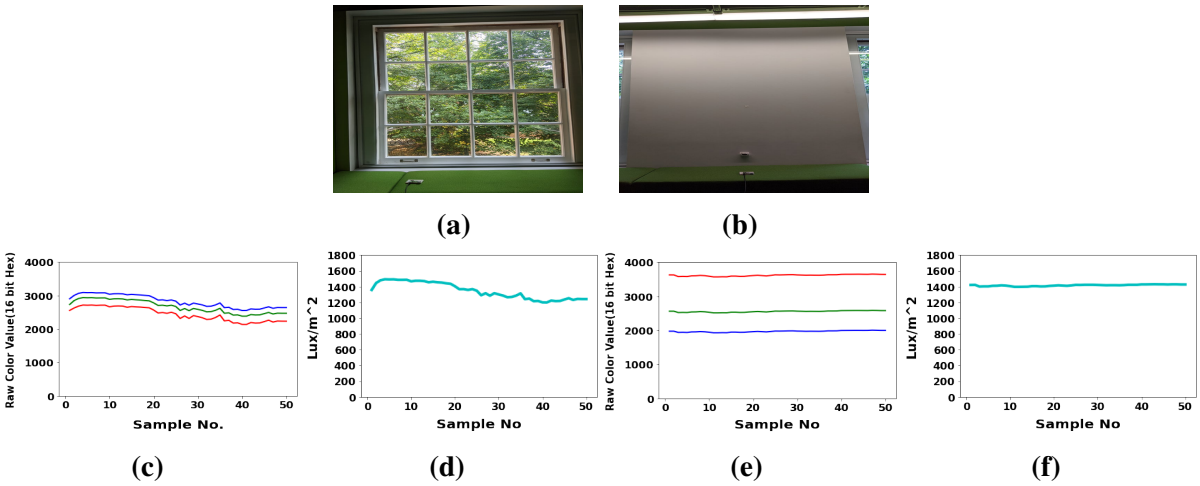


Figure 3.2: Similar Lux with dissimilar RGB readings under different lighting background: Sensor placed under: (a) Natural light, (b) *LED*. Recorded RGB and Lux with Natural light [(c) and (d)] and *LED* [(e) and (f)]. For RGB values: *x-axis*: Number of Samples, *y-axis*: Recorded RGB numbers (in decimal). For Lux intensity: *x-axis*: Number of Samples, *y-axis*: Recorded Lux/m^2

3.0.2 Properties of common indoor bulbs

In this work, we consider three most widely used indoor bulbs with sunlight: Incandescent lamps, Compact Fluorescent Light bulbs (CFLs) and Light Emitting Diodes (LEDs). Radiation is generated through heating tungsten filament for incandescent bulbs. CFL mostly offers ”cool white light” and spectrum exhibits certain spikes during the startup phase [35]. Led delivers radiance over a wide band of wavelengths, like soft white (2700K-3000K), cool white (3100K-4000K), daylight (5000K-6000K) etc. Emissive surfaces of LEDs are highly-concentrated, illuminance of which can be 1000 times higher than recommended level [36]. Although sunlight covers the broadest spectrum, its nature is dynamic, intensity and color components of light (wavelengths) change with the time of day, time of year, the weather and the location on earth. Figure 3.3 illustrates the characteristic RGB variations of various source types as recorded by LPCSB. Different types exhibit distinct signature patterns, which can serve as indicators of their respective classes.

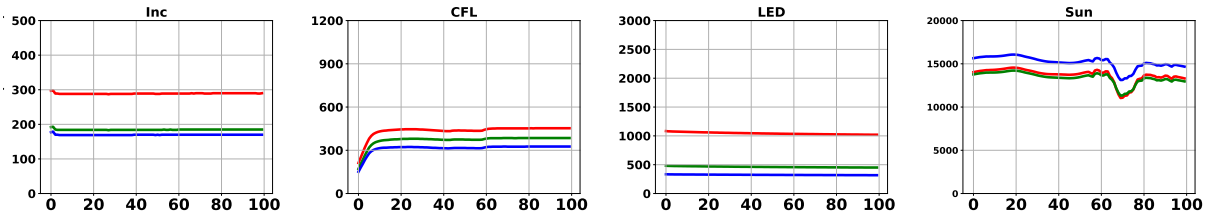


Figure 3.3: 100 samples of sensed RGB values for each type of light source (x-axis:sample no., y-axis: equivalent decimal value): Incandescent (”soft white”, 40 W), CFL (”natural daylight”, 13 W), Led (”soft white”, 9 W), Sunlight (open, 12:45 pm)

3.1 Related works

Source classification techniques till date has been primarily relied on spectrum data from mini-spectrometers. *C12666MA* mini-spectrometer from Hamamatsu electronics has been the most favored which costs around $400, operates on 4.75-5.25 V range and consumes power around 30mW. Mini-spectrometers from *Pasco* can operate on wireless mode, but again costly (around $450) for multi-location mass deployment. Low-cost and portable spectrometer using CMOS-based sensors was designed which is able to detect wavelengths in a range from visible to NIR region. Named *AvaSpec-Mini2048CL spectrometer*, different types of electric lights, along with natural light source were chosen for capturing class variation and MLP model was used for data reconstruction. Prediction errors were calculated for different indoor and outdoor conditions after comparing with *Wavego* [37]. Fernandez [38] utilized RGB information from *TCS3414CS color sensor* and *ADJDS311 color sensor* to classify various artificial sources (34 LED, 16 incandescent and 6 fluorescent sources) and selecting a model estimation of Color Rendering Index (CRI) and Correlated Color Temperature (CCT). Ma, Bader and Oelman [31] did similar kind of research with *TSL2561, ISL29125* color sensors, *AM1815CA, POW11D2P* solar cells and *USB2000+* spectrometer, where sensor data for Halogen, Fluorescent, LED and Incandescent bulbs were collected via USB interface and I-V tracers and KNN, SVM and Decision tree algorithms were utilised for classification for the most part. It has been displayed that even with higher intensity interference from other sources, ML based approach can typify sources with only 62.5% outside training specimens.

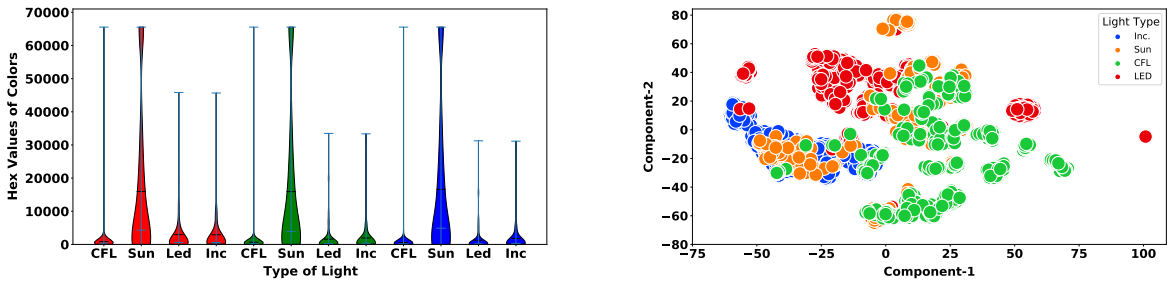


Figure 3.4: Violin plot depicts RGB values of different classes of light source lie in common range which makes it difficult to categorize based on cutoff intensity (Mean and Median values are marked with white and black lines (left), 2D tSNE plot of recorded RGB information reveals linearly inseparability of source clusters (right))

3.2 Methodology

We utilise LPCSB to collect color sensitive information from from 27 different bulbs (9 from each of LED, Incandescent and CFL) for acquiring both inter-class and intra-class variation of RGB values. Raw data measurements taken from the color sensor reveal the amount of red, green, and blue components that compose the unfiltered light. To ensure optimal performance, the sensor is positioned facing the light sources, although the angle of capture is not expected to affect classification accuracy since it does not alter the temporal patterns [39].

To achieve true nature of each light by minimizing influence from other sources, we decide to carry out all the measurements (for artificial bulbs) in a dark room. For collecting sunlight data, we expose sensor to sun in diverse conditions and scenarios, which includes taking data from sunrise to sunset, during heavy rainy, foggy and drizzling days. Inconsistency of sunlight RGB information may also derive from contrasting indoor conditions (location, window glass material, with and without blinds etc.). To accommodate them into our training set, we collect sunlight data in different buildings and also in various corners of a building.

In practice, electromagnetic light waves experience reflections from nearby structures. Finally, sensed complex signal, deriving from contrasting scenarios and mixed with direct and indirect components, simply do not follow inverse square law of radiation and generates irregularity in RGB values. To acknowledge magnitude variability and irregularity of RGB features based of sensor placement, we capture artificial light data at five different distances.

For analysis, we collect 500 samples for each observation. To determine optimal sample size for classification, we divide our collection window size from 10 samples up to 125 samples. Figure 3.4 shows RGB distribution of all the sources dealt in this study. As discovered, unalike sources share common RGB spectra and magnitude, which turns it problematic to differentiate solely based on RGB threshold. t-SNE visualization of RGB values also reveals the fact that dissimilar light sources are linearly inseparable. That's why we investigate multiple Machine Learning(ML) and Neural Network(NN) algorithms to distinguish each type of source. ML and NN models are independent of feature magnitude after scaling and capable of non linear classification. As RGB signals contain resemblance with image data (both are primarily three channel information), we inspect both feedforward Multilayer Perceptron Models (MLP) and Convolutional Neural Networks, with 1-D and 2-D filters (CNNs) for categorization. As sensed data is time series based, we also include Long Short-Term Memory (LSTM) network for typifying. There can be a vast number of neural network architectures, each with a large set of variable parameters. For the selection of NN-based networks, we aim to ensure that the networks achieve high accuracy while maintaining manageable complexity for efficient execution. We experiment with several fundamental neural network parameters *(such as the number of layers, optimizers, dropout rates etc.)* and report the configuration that performs best on our dataset. The complete list of classifiers studied is shown in Table 3.1.

After training and fine tuning our chosen classifiers with controlled environment data, we record performance of each classifier based on different size sample window and sensor placement. While training, we scale, normalize and divide the balanced dataset into training, test and validation sets (80%, 10% and 10% respectively). For better evaluation and to ensure representation from each group, we imply stratified 10 fold cross validation by tuning classifiers to their best hyper parameters using *Gridsearch*. For identifying primary source in a multi-source environment, we blend RGB values light sources and observe whether our classifiers can identify the primary source. While mixing, we make sure that the RGB values from second/interfering source never goes past values from primary source, as classifier is expected to determine the major contributor between/among sources. In previous work like [31], only constructive interference has been considered as the consequence of overlapping. For further investigation, we place two light sources near the sensor and compare the resultant with simple theoretical addition. We find that they differ by a large margin, both in RGB and in lux domains (figure 3.5).

Table 3.1: Methods of classification

Method	**Acronym**	**Tuned Par./ Model Par.**
Decision Tree	DT	*criterion, max depth*
Random Forest	RF	*max depth, max features, min samples leaf, min samples split, n estimators*
Gaussian Boost	GB	*learning rate, max depth, min samples leaf, min samples split, n estimators*
Naive Bias	NB	*var smoothing*
K Nearest Neighbor	KNN	*metric, n neighbors,weights*
Logistic Regression	LR	*C parameter, penalty*
Support Vector Machine	RBF (SVM-Rad) Linear (SVM-Lin) Polynomial (SVM-Poly)	*C parameter, gamma*
Multilayer Perceptron	FNN	*No of layers:7, Dropout:20%, Activation:relu, softmax, Optimizer=SGD, Loss = Categorical cross-entropy*
Convolutional Neural Network	CNN-1D CNN-2D	*No of layers: 6 (1-D)/7 (2-D) , No. of filters: 64/32(1-D),64/32/16 (2-D), Kernel Size* $= 2 \times 2$ *(1-D),* 3×3 *(2-D), Padding=same, optimizer= Adam, Dropout:20%*
Long Short Term Memory	LSTM	*No of layers: 4 , output dimension= 50,optimizer= Adam*

Based on the phase difference resulting from positioning of both sources at sensor point, a numerous blending ratio is possible. Variety of mixture represents blending of constant positioned sources at different sensor placement or positioning of sources at different locations, sensed at the same spot. However, highest possible deviations are amalgamation of identical and opposite phases. As our goal was to testify our classifiers, we add only those extreme cases that can lead into inaccuracy with the highest probability.

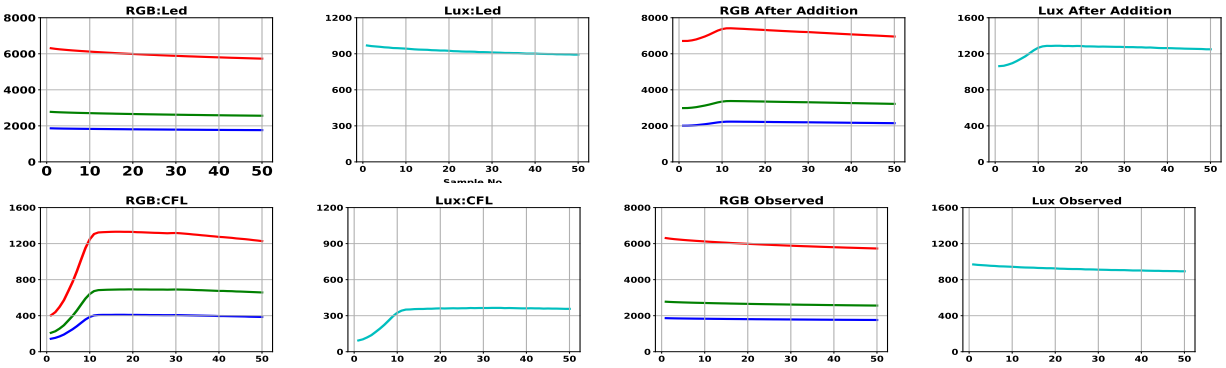


Figure 3.5: RGB observations (x-axis: sample no., y-axis: raw RGB values) along with lux intensities (x-axis: sample no., y-axis: lux/m^2) from multiple sources, LED was set as primary and CFL as secondary source. Observed outputs deviate from simple addition

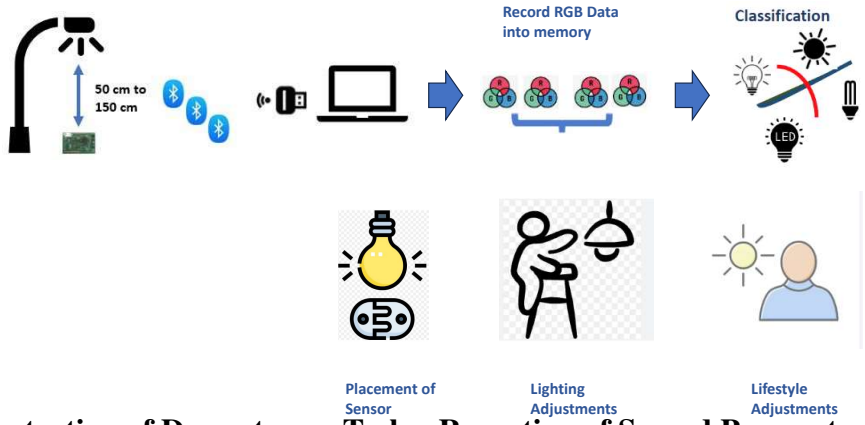


Figure 3.6: Illustration of Downstream Tasks: Reception of Sensed Parameters from Indoor Light Sensors, Analysis, and Subsequent Adjustments Based on Outputs

3.2.1 Downstream Tasks Utilizing the Classifier Output

As previously discussed, the LPCSB senses environmental data and transmits it to a remote PC. Upon receiving the information via a BLE receiver, the remote platform temporarily stores the data before executing the classification and saving the results for subsequent analysis. Based on these outcomes, users can take various actions, such as replacing the fixed-point light sensor to enhance performance, adjusting the light type to suit specific needs, or modifying exposure to natural light throughout the day to promote well-being (figure 3.6)

3.3 Evaluation

We analyze prediction accuracy in different background and record mean values of classifying accuracy (with standard deviations). As initial accuracy was high with only using RGB data, we discard clear value, lux intensity, or color temperature readings for classification. Although artificial lighting landscapes do not change very often in indoor landscapes, classifiers should be robust enough there to classify under inexperienced screenplays with factual mishaps. We start with the ideal scenarios to fix parameters for indoor deployment and then progress with evaluating non-ideal incidents with the settled values. We illustrate the findings mainly with violin plots, which depict distributions of numeric data through density curves on both sides of the mean value. Accuracy values exceeding 100% in those plots were trimmed.

3.3.1 Prediction in known scenario

Left plot of Figure 3.7 illustrates variability of mean accuracy for different classification techniques with varying number of samples. Observation reveals that the classification accuracy **is not linearly proportional to the number of observed RGB samples.** The best average result is achieved with 50 samples, although the average accuracy with 10 and 25 samples was also close to 90%. Right plot of Figure 3.7 represents a comparative analysis among classifiers, trained with

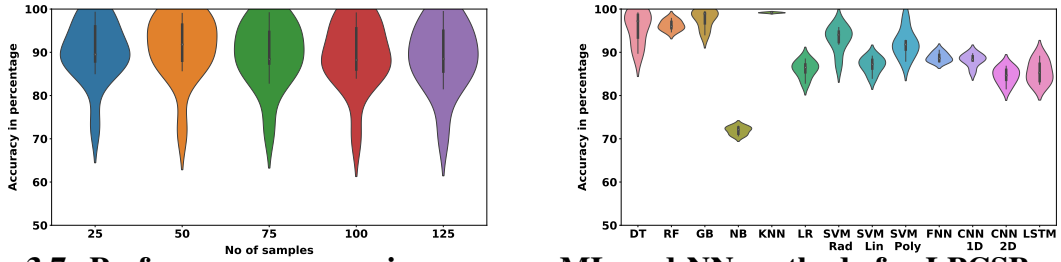


Figure 3.7: Performance comparison among ML and NN methods for LPCSB at indoor where training windows were varied from 10 to 125 samples: 50 samples triggers the best performance and KNN was the best performer (left), KNN performs the highest among classifiers (right)

different sample sizes. It uncovers that overall performance of ML algorithms is better than NNs at ideal and known scenario. There can be a number of reasons for that, which includes dataset nature and size. However, K-Nearest Neighbor (KNN) triumphs for generating highest and most consistent accuracy among all. To investigate why KNN outperforms other classification methods, we plot the

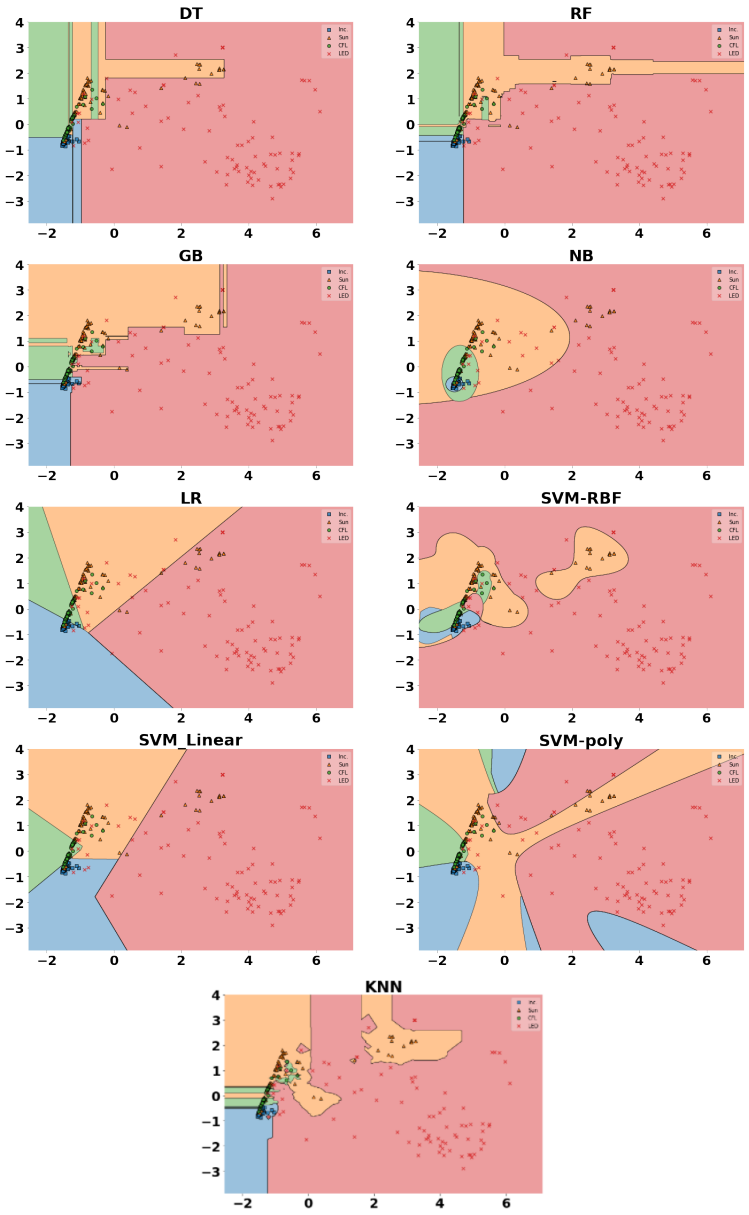


Figure 3.8: Decision Boundaries for ML classifiers

decision boundaries of various ML methods against KNN. The plots reveal that similar types of bulbs form clusters in the RGB space, and KNN's complex, non-linear decision boundaries handle these clusters more effectively than other methods (Figure 3.8). It is also notable that although the collected data is time series based, accuracy of CNN is higher than LSTM. This occurs when the

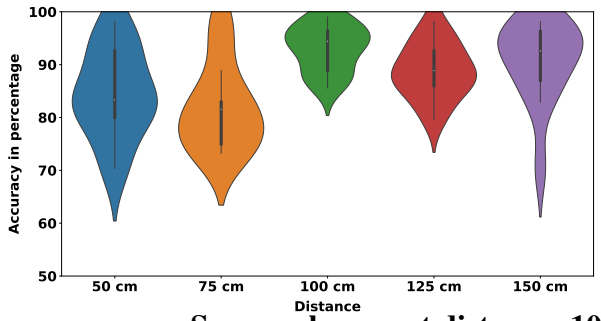


Figure 3.9: Variation in accuracy vs Sensor placement distance: 100cm yields the maximum mean accuracy

data has strong local patterns, and classification depends more on these features than on long-term temporal relationships.

To carry on with KNN, our goal is to find the sweet spot for balancing number of samples with accuracy. After observing KNN accuracy with varying samples window size, we conclude **25 samples window exhibits the combination of highest accuracy and lowest standard deviation.** We continue our analysis with 25 samples window for both capturing the transient and stable state of radiation and accommodating minimum number of samples at packet loss scenarios.

3.3.2 Placement of sensor

Our goal was to observe if we had the liberty of deploying sensor at any distance from the source indoor, where should we place it to achieve maximum identifying accuracy. We place LPCSB at 5 different distances for Inc, CFL and LED bulbs, starting from 50 cm to 150 cm to observe whether placement of sensor plays any role in classifiers' performances. Our analysis shows that **placing sensor at 100 cm can detect the background source with maximum accuracy** (figure 3.9). Now, we focus on non-ideal situations with known sources/scenarios and testify classifiers performance by setting up 25 samples window length. For investigation, as before, we include 80% examples in our training set to familiarize our classifier and 10% each for validation and test sets.

3.3.3 Classifying the prime source in a multiple source environment

To fabricate multi source environment, RGB values from second source was mixed at different amount, varying from 20% to 80%, of the intensity of the primary source. Highest possible

deviations were included for analysis (through addition and subtraction of RGB values of primary and secondary sources) and mixing signals from all possible combinations (LED/CFL, LED/Sunlight, Sunlight/LED+CFL etc.). Our study reveals although classifiers accuracy declines with increasing mixture ratio but overall performance do not fall significantly in multi source environments (figure 3.10).

3.3.4 Identifying in presence of random noise

In real world, nearby elements can act as a noise source by reflecting particular component of light which can escalate or descend the sensed values. However, by placing RGB reflecting elements nearby, we find that finally recorded value contain very were small interference. We vary the influence randomly from 0% to 5% of maximum RGB values (without noise) and enlist the performances (figure 3.11). As recorded, accuracy decreases with increasing intensity of perturbations. All inclusively, NN based classifiers can withstand turbulence better than ML based classifiers. Random forest performs best among ML algorithms (mean accuracy 84.46%) where accuracy score of KNN was close to that (mean accuracy 83.17%). This reflects that NNs can generalize better and become less sensitive to small disturbances than ML based classifiers.

3.3.5 Detection Precision in smart environment

For source detection in smart environments, we vary the on/off duration of sources. In addition to creating peaks and troughs, the incident also alters the minimum values recorded for each color component, depending on the length of the off period. For that, we vary the lowest value cutoff

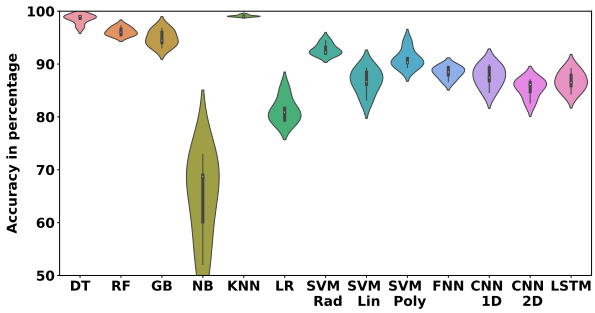


Figure 3.10: Overall accuracy at multi source environments where secondary source intensities were varied in between 20% to 80% of the primary

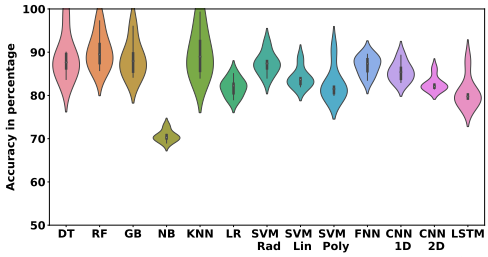


Figure 3.11: Overall accuracy based on randomly varying RGB values in between 0% to 5% of the major source intensity

points from 20% to 80% of the maximum value to represent different off-periods. As observed, the final RGB signal patterns showed significant shifts from the initial pattern based on the threshold (Figure 3.13(left)).

Now evaluating the performances to identify the altered patterns, we inspect that LSTM is the best performer (Figure 3.12). Random switching can totally disrupt the local features which greatly impacts ML based classifiers' accuracy. LSTM excels at capturing intricate temporal dependencies and adapting to dynamic patterns. This makes them more robust to unexpected fluctuations and sudden changes in the time series data.

If we enlarge the accuracy in KNN case based on threshold, we can see that accuracy of classification decreases with lowering threshold values (figure 3.13 (right)). So balancing threshold with KNN is a pre-requirement for desired accuracy in smart environments.

3.4 Real-world Deployment

Here, we deploy our board in real world settings. For classification, we single out KNN as our classifier, for exhibiting the most consistent performance in all scenarios and trained it with all ideal/non-ideal examples from controlled atmosphere. We conduct 3 experiments at 3 different test

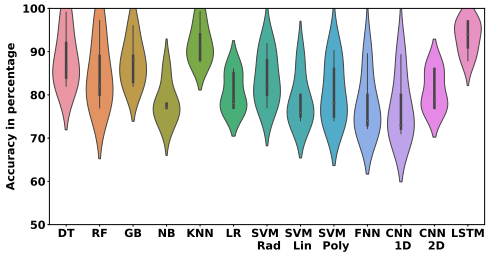


Figure 3.12: Overall accuracy in primary source detection at smart environments

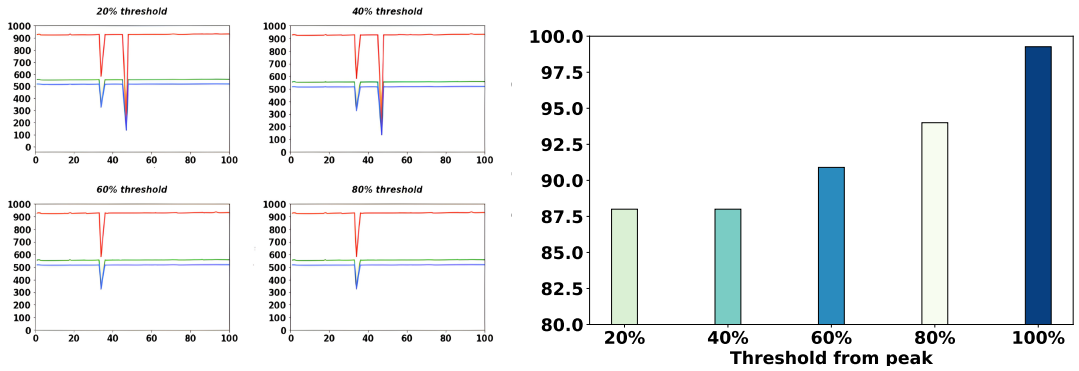


Figure 3.13: Different patterns created for different lower thresholds (left, x-axis: time samples, y-axis: dec values), Declining detection accuracy of KNN classifier was observed with lower cutoff settings (right)

beds: (1) Household, (2) Lab environment-1 and (3) Lab environment-2. All tests were performed with completely unknown artificial bulbs. Experiments included single source, mixed light source and arbitrary switching of bulbs scenarios (figure 3.14).

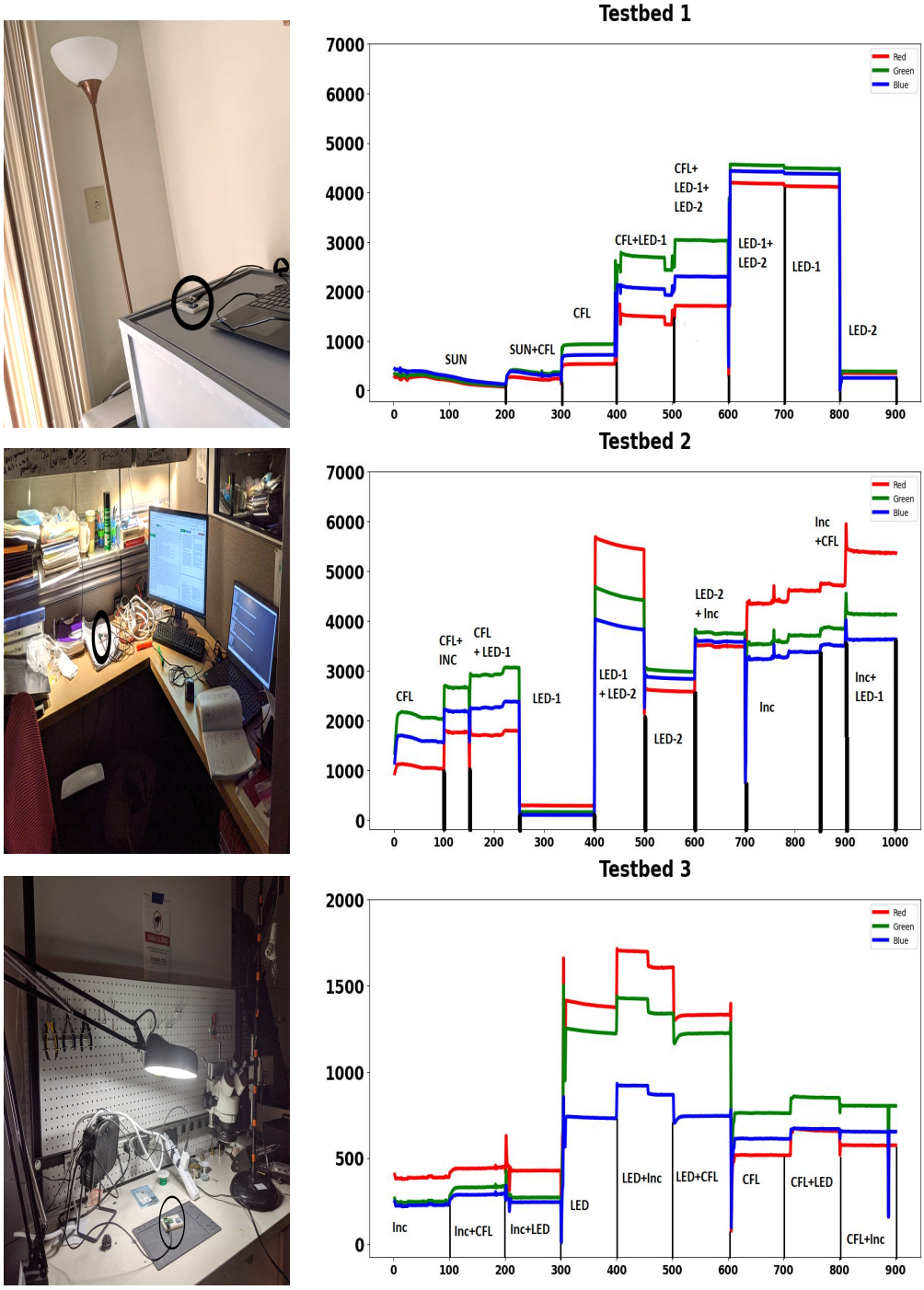


Figure 3.14: RGB signals captured in test-beds, circles point placement of LPCSB (x-axis:sample no., y-axis: recorded decimal value)

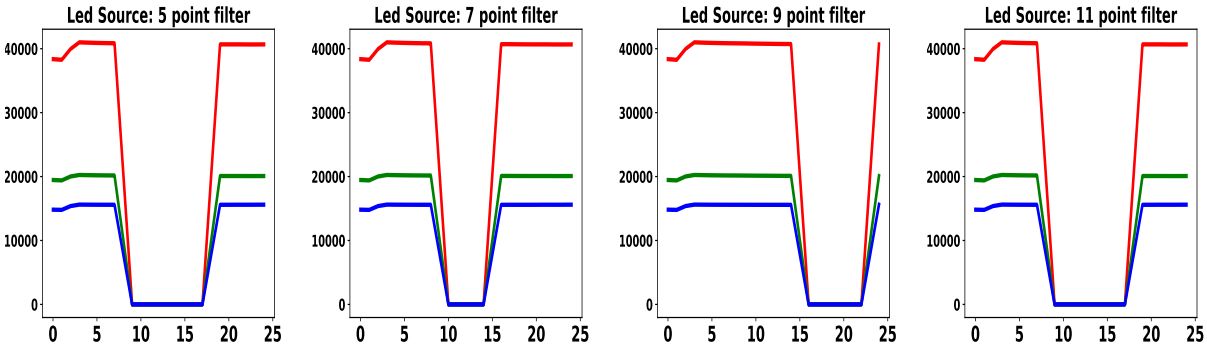


Figure 3.15: Applying filters of different sizes on a LED source to capture fluctuations in a 25 RGB sample window: (x-axis: sample no., y-axis:hex value)

3.4.1 Analysing misidentified examples

After analysis, we observe that classification accuracy has been degraded unexpectedly. Classifier got confused during few transition events. We also observe that the faulty predictions were not common for any particular light. Moreover, for artificial lamps, a single source at different distances have been typified as different classes. While detecting RGB spectrum of sunlight during sunrise and sunset, classifier has been misguided.

3.4.2 Addressing Transient Patterns

As explained, the best performing classifier in controlled experiments failed to identify transient patterns in realistic testbeds. These transient patterns are generated due to multiple reasons, which includes: **sudden interference in between source and the sensor or switching of lights**. These fluctuations during data acquisition may occur arbitrarily and for unknown duration. When our sensors records zero RGB values, it is practically impossible to detect the source type. But if it senses non-zero values even for some duration, we may utilize that information for source classification.

To familiarize our classifier models with those events, we have designed filters of different window sizes and randomly implemented them to the examples from controlled environments (shown in figure 3.15). After generating these examples, we add them into the training set. These examples are expected to familiarize our classifiers with transient patterns and increase the real-world accuracy.

3.5 Representing Unseen Light Sources with Synthetic Dataset

With limited amount of data, machine learning models tend to over-fit and become problematic for non-linear classification. However, at the same time, it is unrealistic to include all the light sources available in the market in our training set.

To familiarize our classifier with unseen examples, we decide to generate synthetic dataset based on current data distribution. While there are various methods to generate synthetic data, our time-varying RGB dataset includes both static (e.g., constant values during LED/ screensaver mode) and dynamic (erratic variations during sunrise-sunset/video on screen) features, which need to be considered. To tackle the complexity of generating time series data with intricate temporal dependencies, we employ the TimeGAN architecture [40] and generate equal number of synthetic examples of captured data and based on current data distribution. This approach captures irregular and random fluctuations, as well as constant attributes, providing more challenging examples for models compared to simpler methods like autoregressive models.

There are different methods to verify whether the GAN generated synthetic examples actually represents original data distribution. This includes visualization techniques, statistical tests (measuring wasserstain distance or KL divergence), feature space similarity (measuring Fréchet Inception Distance), classification performance, log likelihood estimation and so on [41]–[43]. In our case, as our data is Sensor based time series observations, we pick PCA from visualization techniques and TSTR (Train Synthetic, Test Real) metric for evaluation [44], [45].

Figure 3.16 demonstrates distribution of first two principal components of real and synthetic examples generated using TimeGAN. Comparing the principal components of the generated and collected datasets showed that the generated data closely resembles the original data distribution. As observed, the synthetic data did not maintain equal diversity across all source types. The similarity between original and synthetic data was highest for LED bulbs, indicating stronger consistency in that category. In contrast, CFL and natural RGB examples exhibited greater experimental variability, a trend that was also reflected in their synthetic counterparts. The TSTR accuracy was found **90.4%**. However, whether the generated data set actually represents any real world light source example or not, we verify that on realistic testbeds.

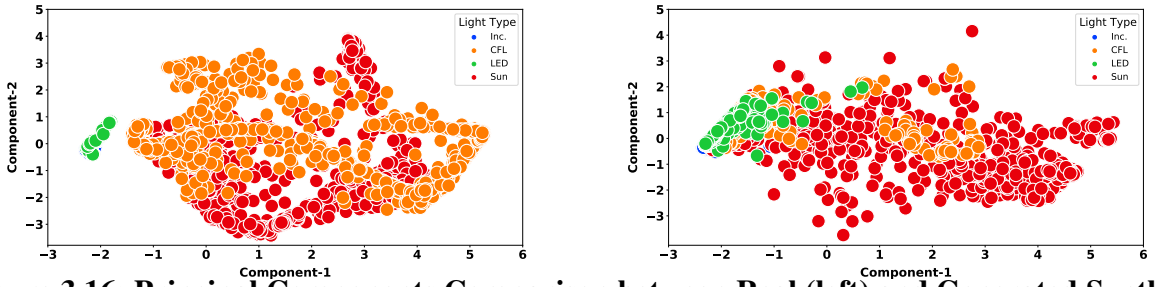


Figure 3.16: Principal Components Comparison between Real (left) and Generated Synthetic Examples (right) using TimeGAN

3.6 Retraining with Augmented training set

Now, we retrain our KNN classifier with extended training set and re-record the accuracy. Our investigation reveals that the performance of the KNN classifier has been significantly improved (Figure 3.17), both in identifying new sources and in observing sources that involve random switching (Figure 3.18). This exhibits that the augmented dataset correctly represents unseen sources and the transient patterns.

Even with elevated training, we observed examples that were failed to get correctly identified. A few of them have been listed below (figure 3.19). Again, no single pattern of mis-classification was discovered.

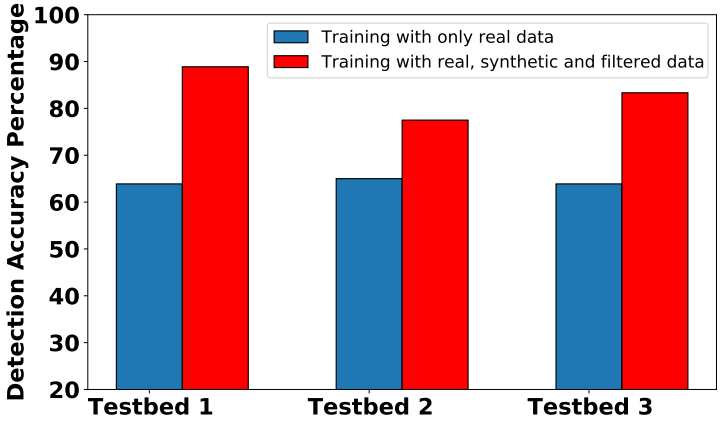


Figure 3.17: Accuracy reveals KNN with extended training set exhibits superior performance in unfamiliar environments

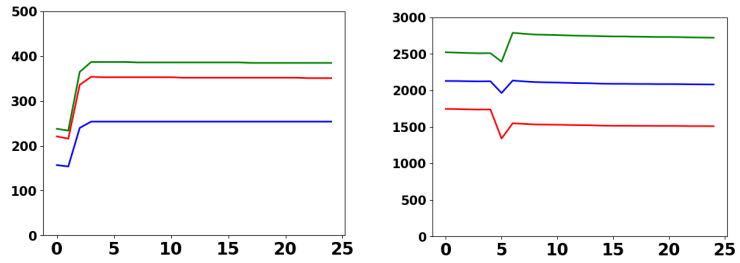
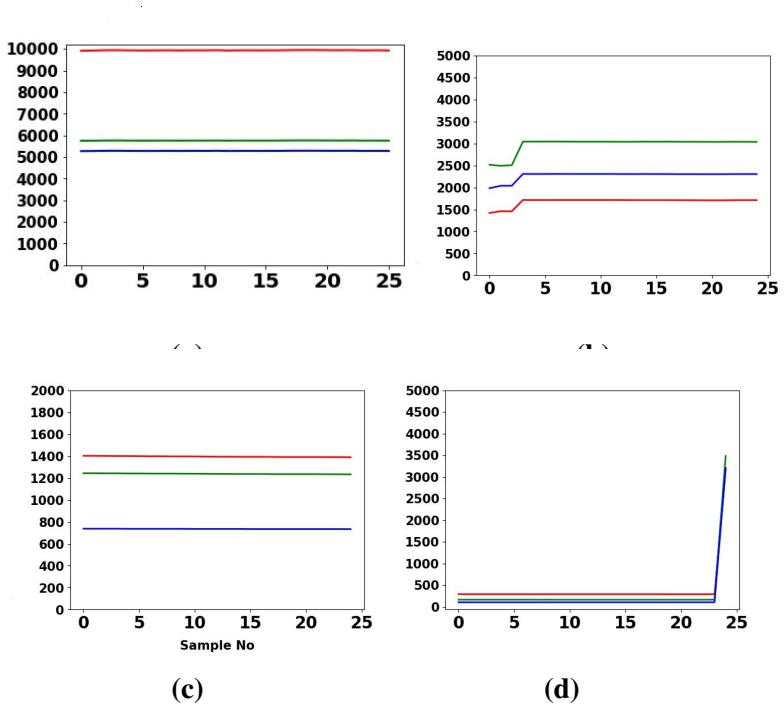


Figure 3.18: Correct Predictions with extraneous training set (x-axis: sample no., y-axis:hex value) : During Switch-over (left):Prev-Led, Now-CFL. During random movement (right):Prev-CFL, Now-Sunlight



(c) **(d)**

Figure 3.19: Miscategorized examples with Incandescent (”work white”, 150 W),Led (”warm yellow light”, 40 W) and CFL (”T9, 6400K”, 22 W) bulbs (x-axis: sample no., y-axis: dec value): (a) Inc. as Sunlight (b) Led as Sunlight (c) Led as Inc. (d) CFL as Led

Chapter 4

IDENTIFICATION OF SINGLE-SOURCE / MULTI-SOURCE INDOOR LIGHTING ENVIRONMENTS

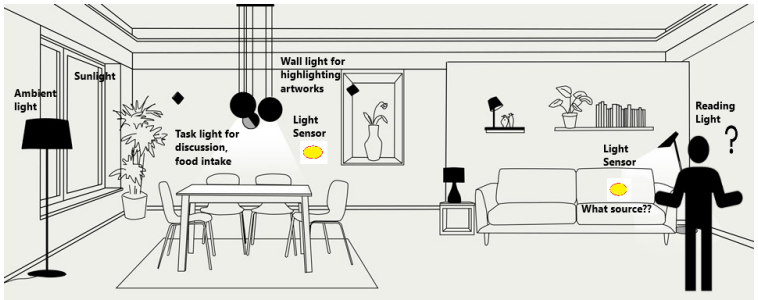


Figure 4.1: Identifying indoor lighting in a Single/Multi Source Scenario

As previously discussed, a significant limitation of current source classifiers is their design, which primarily focuses on identifying a single-source lighting environment, even in scenarios involving multiple sources. This constraint reduces their sensing capability, particularly when it comes to detecting the presence of a specific source type within the background of another or understanding how multiple source types combine to create a specific luminous environment within an enclosure.

4.1 Methodology

In this work, after deployment and data collection under single source environments, we try to recreate multi-source environments through blending sources with different ratios. We divide the collected data set into 10 different classes, including single and multi-class combinations. To mimic multi-class scenarios, we mix signals from different light sources at various ratios where the secondary signal intensity went down up to 20% of the primary.

In a multi-source environment, when nearby sources interact, the resultant depends on different measurements: (a) the relative distance between two sources and (b) the positioning of the sensor.

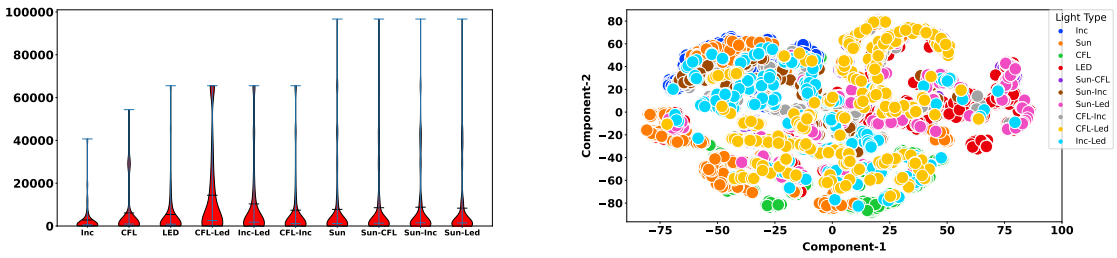


Figure 4.2: Violin plot showing red color component distribution of training data [left] (x-axis:source type, y-axis: recorded values). As observed, different classes generate common readings and cannot be segregated solely based on threshold setting. First two principal components of 2D tSNE plot exhibits linear inseparability of single and multi-type source environments [right]

The captured resultant, $\vec{E_T}$, depends on the relative phase difference at the sensor location.

$$\vec{E_T} = \vec{E_1} + \vec{E_2} + 2\sqrt{|\vec{E_1}||\vec{E_2}|}cos\theta \tag{4.1}$$

where, $\vec{E_1}$,$\vec{E_2}$ are electromagnetic light waves from two different sources and $\vec{E_T}$ is the resultant. θ depends on the path difference. Based on θ, in extreme cases, multi-sources can create both constructive and destructive interference at the sensor locations, where the possibility of misclassification is the highest. In our analysis, we have considered both extremes for classification by adding and subtracting RGB signals from multiple sources.

Figure 4.2 (top) shows the red color component distribution of all samples. As depicted, different sources/scenarios share common values, which were also observed for Green and Blue readings. As a result, classification based on color-magnitude is inaccurate. Two-dimensional tSNE visualization of RGB values depict linear inseparability of classes (Figure 4.2) (bottom).

After collecting sensor data, we have scaled, normalized and divided the balanced dataset into training, test and validation sets (80%, 10% and 10% respectively). For the classification of the captured non-linear information, we employ the same set of classifiers discussed in the previous chapter to accurately identify the primary source. For better evaluation of the built model and to ensure representation from each group, we have implied stratified 10-fold cross-validation by tuning to their best hyperparameters using *Gridsearch*. After collecting the results, we analyze the examples where our chosen architecture went wrong in identifying the source type and how they differ from correctly predicted examples.

4.2 Evaluation

We analyse prediction accuracy in different backgrounds and record mean values of classifying accuracy (with standard deviations).

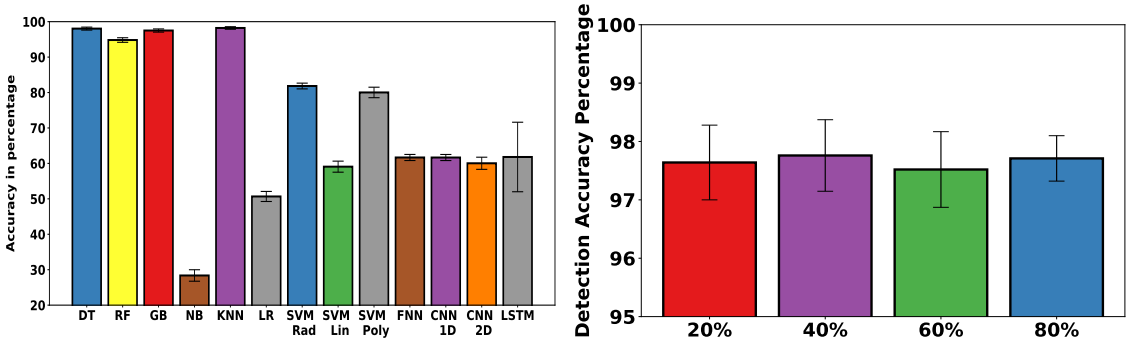


Figure 4.3: Comparing accuracy among different ML and NN architectures(left), Accuracy with KNN after mixing two sources at various proportion reveal disassociation of accuracy with mixing ratios (right)

4.2.1 Performance of classifiers

Figure 4.3 illustrates variability of accuracy for different classification techniques. As observed, overall performance of ML algorithms is better than NNs in known scenario. The best result with maximum average accuracy (upto 98.2%) with minimum deviation was recorded with KNN. Now we observe whether blending ratio has any impact on the accuracy. As seen in Figure 4.3 (right), accuracy is not proportional to mixture ratio of multiple sources.

With tuned ML classifiers, we then examine how each classifier performs at smart on/off environments and at identifying sources outside training set. As seen from Figure 4.4, KNN performs the best at classifying at smart scenario whereas Random Forest (RF) is the highest performer at identifying unknown sources.

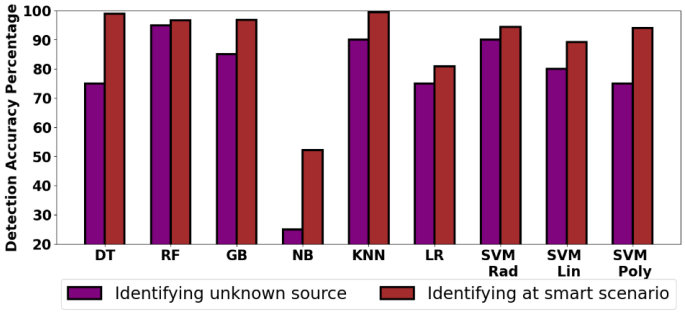


Figure 4.4: Variation of accuracy for tuned ML classifiers at smart env. and with unseen sources

Chapter 5

DIMENSION REDUCTION FOR INDOOR SOURCE CLASSIFICATION

Figure 5.1: Identifying indoor lighting in a Sensor Crowded Scenario

Light monitoring may be intended only for certain business hours during a day *(like monitoring indoor ambience at commercial stores or marketplaces)*, where sensors can afford higher memory, power and dimension. Even in such places, offloading sensed parameters to distant devices (*like a distant PC or server*) is always encouraged, as it allows deployment of available light sensors in the market that have limited storage to store few observations, transfers major computational and memory intensive classification task on the receiving side, and facilitates long term data storage and retrieval [46].

Enclosures like commercial stores or other sorts that are often overcrowded with multi-type sensors, packet loss scenario is common due to network congestion and communication bottleneck [47]. As a result, off-board classification becomes challenging. Advertising fewer informative packets there is advantageous, by bringing down number of successful receptions needed for classification. However, appropriate methods to minimize on air traffic should be investigated, as it may impact

the selection of sensors and classification accuracy. For efficient off-board identification by minimizing the required number of advertisements, we have *investigated* different dimension reduction methods with the necessary number of principal components. Dimension reductions were accomplished using with and without original data reconstruction approaches. We *compare* them based time, power and memory requirements and *analyse* the performances of all the different techniques both in ideal and real-world scenarios.

5.1 Dimension reduction

With the current methodology, the classification process begins only after receiving 25 successful samples. At a sampling rate of 5 samples per second, this translates to a mere 5-second timeframe. During this short duration, the variation in the parameters sensed between consecutive samples is minimal. This limited variability motivates the adoption of **Dimension Reduction** techniques for data efficient Indoor Light Source Classification [48].

25 samples with 3-channel information can be alternatively considered as 75-dimensional data. With dimension reduction technique, these high dimensional data can be reconstructed using fewer dimensions, with necessary information packed inside. With modified dimension, compressed RGB data have elevated robustness, and can be advertised even in a single packet with repetition if necessary, which significantly truncates the operational power budget. Although our current BLE device is incapable of such computation, common IoT based edge devices in market have additional memory and power capacity to run dimension reduction algorithms on a chip.

In this work, we analyze classification by compressing (dimension reduction) sensed RGB data using different compression techniques, record their accuracy, and compared the result with the uncompressed method (Figure 5.2). Two different approaches were analysed for classification with compressed information, as shown in Table 5.1. For dimension reduction, we choose six different

methods: Principal Component Analysis (PCA), Truncated Singular Value Decomposition (Trun-SVD), Linear Discriminant Analysis (LDA), Kernel-PCA with cosine function (KPCA), ISOMAP and tSNE. Our goal is to select the best classifier among all scenarios. We start with taking only two principal components with method 1, apply our best performing classifier KNN for classification and compare accuracy with familiar sources, smart on-off scenario and with

Table 5.1: Approaches for Dimension Reduction

Method 1	***Method 2***
Step 1: Transform training data with reduced dimension, tuning ML models with training data and storing classifier model at receiver	*Step 1*: Storing classifier model at receiver after tuning with original training data (25 samples consisting RGB info)
Step 2: Storing specific number of RGB samples, transforming stored data with reduced dimension and advertising	*Step 2*: Storing specific number of RGB samples, transforming stored data with reduced dimension and advertising
Step 3: After receiving advertised packets, we feed the input values as input to the classifier	*Step 3*: After receiving advertised packets, first we reconstruct data to get back the original dimension, then we feed values as input to the classifier

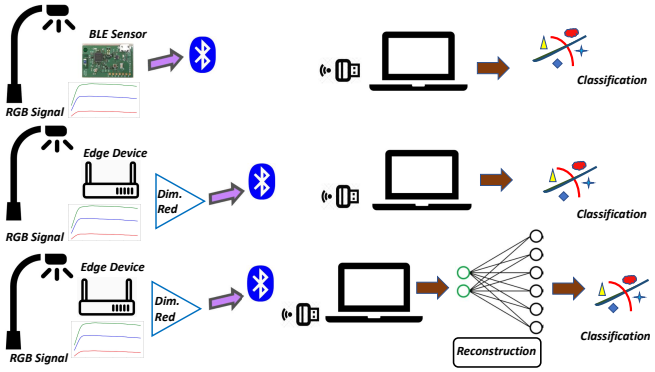


Figure 5.2: Different classification approaches: Without dimension reduction with our BLE sensor (top), with reduction by method 1 (middle) and method 2 (bottom)

unfamiliar sources. We select PCA from linear and Kernel PCA from non-linear technique and consider dimensions up to 10 principal components (based on Figure 5.3). We then classify with reduced dimension utilizing *method 1* (PCA/KPCA) and *method 2* [PCA-Reconstructing from PCA (RPCA)/ KPCA-Reconstructing from Kernel-PCA (RKPCA)].

Our tests with different methods and various scenarios exhibit classification accuracy varies with varying principal components (Figure 5.4).

Finally, we deploy our sensor at two different real world testbeds that include single/multi source (both with similar and dissimilar types) scenarios with completely unfamiliar indoor bulbs, random switching from one type to another and human movements. Based on the results in all scenarios in

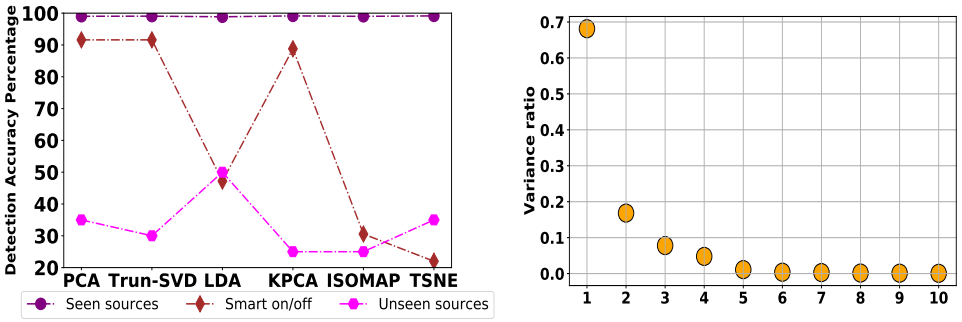


Figure 5.3: Comparing reduction techniques performances reveals superiority of PCA for linear and KPCA for non linear classification (left), PCA over collected data reveals overall variance lies within first few principal components (x axis: no of principal components, y-axis: variation ratio) (right)

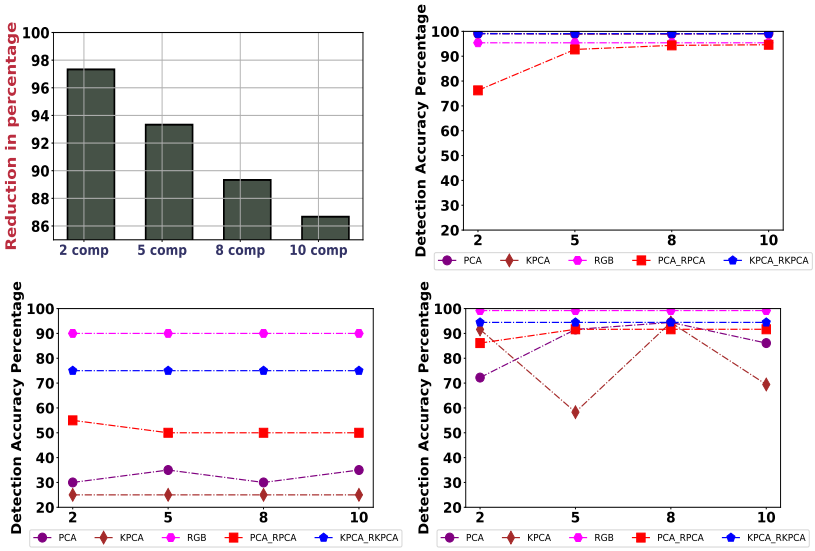
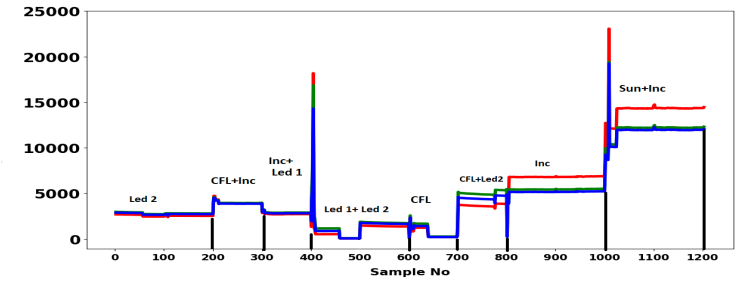


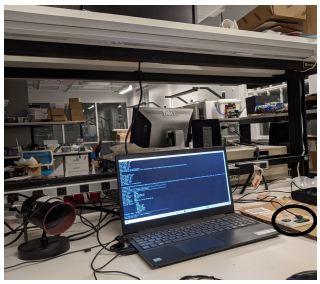
Figure 5.4: Reduction of on-air traffic for advertising only principal components compare to advertising 25 RGB samples (x-axis:no. of principal components, y-axis: Reduction Percentage), Comparing accuracy: familiar (top right), unfamiliar (bottom left) and smart on/off scenario (bottom right)(x-axis:no. of principal components, y-axis: accuracy)

Figure 5.4 and classification with minimal information, we consider 2 principal components with PCA, KPCA, PCA-RPCA and KPCA- RKPCA and apply KNN algorithm for classification. We record accuracy with/without reduction methods (Figure 5.5). As seen, after real-world deployment, the classification accuracy degraded significantly from 98.22% down up to 77.5%. Again, incorrect predictions could not be specified for any single/multi source type and noticeably, some of them occurred during switch overs/movements, as the signal patterns were significantly different at those instances. However, KPCA-RKPCA accuracy at both cases were comparable to accuracy with 25

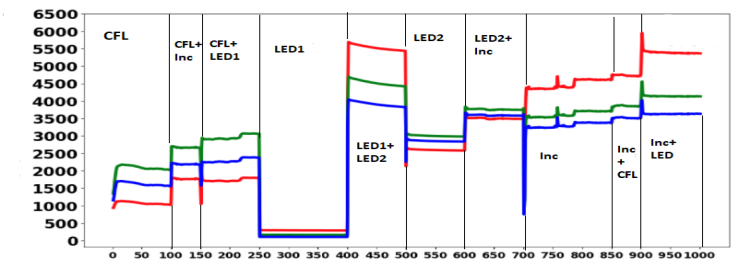
(a)



(b)



(c)



(d)

RGB	*PCA*	*KPCA*	*PCA-RPCA*	*KPCA-RKPCA*
83.33%	41.67%	66.67%	58.83%	83.33%
77.5%	40.0%	60.0%	52.5%	77.5%

(e)

Figure 5.5: RGB readings at two different testbeds (circle point placement of sensor). Recorded classification accuracy reveals performance of KPCA/RKPCA is comparable to RGB value-based classification.

RGB samples.

For general comparison, we run dimension reduction on two different edge devices. Along with recording the highest accuracy, KPCA-RKPCA method demands the highest time, memory and power to operate (Figure 5.6). Compare to PCA, KPCA constructs a full $n \times n$ kernel matrix over n data points to allow for nonlinear relations, which demands extra resources for execution [49].

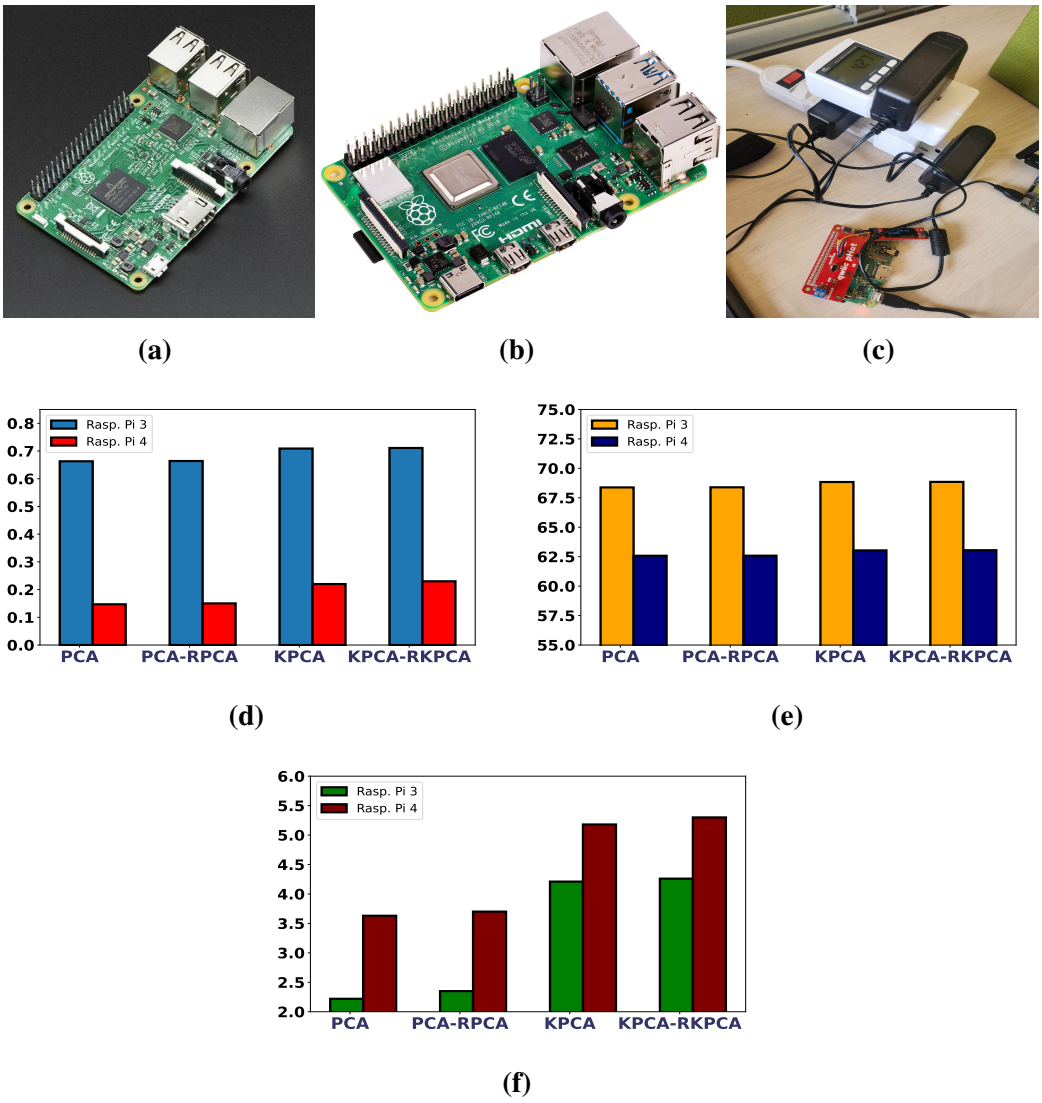


Figure 5.6: Running dimension reduction methods on raspberry pi 3 (a) and pi 4 (b). Measuring power intake while running (c). Overall Comparison exhibits KPCA-RKPCA method requires the highest time (c) [y-axis= seconds], memory (d) [y-axis= MB] and power (e)[y-axis=Watt] for execution

Chapter 6

DATA EFFICIENT SENSING FOR DAYLONG LIGHT EXPOSURE ANALYSIS WITH EVENT DETECTION AND LOSSLESS COMPRESSION

A significant limitation in energy-efficient daylong source exposure analysis lies in the challenge of continuously sensing the constant lighting environment over extended hours for classification. Figure 6.1 illustrates an example scenario involving an indoor light sensor placed near an inhabitant. The sensor records the individual's daylong source exposure by transmitting the surrounding environment's color components and illuminance levels for source classification. At indoors, inhabitants typically remain stationary for extended periods, where the lighting conditions tend to be static. As seen from the Figure 6.1, with such a static setting that lacks intelligent processing, the sensor continuously transmitting the similar parameters and classifying the same source repeatedly.

When the goal is to identify time-specific source exposure throughout a day, this approach proves to be resource-intensive and introduces data redundancy, complicating further processing. This becomes further complicated in scenarios where sensors operate under energy constraints or where replacing power supplies is difficult due to limited accessibility. Furthermore, modern indoor sensors provide low-power operation combined with a degree of intelligence, which remains underutilized when limited to a simple sense-and-advertise approach without incorporating any on-device processing.

In contrast, within a *smart* framework, the sensor can detect the initial lighting scenario and pause transmissions until a source switch-over occurs. It then communicates only the critical information required to identify the new source, eliminates insignificant sample values for characterization, and

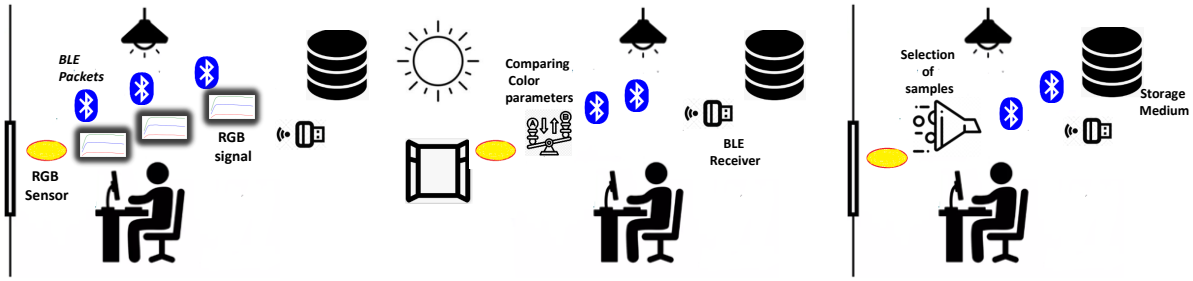


Figure 6.1: Overview: Sending similar packets under constant lighting (left). Opening room window creates new lighting environment. Sensor first senses the event and transmits packets accordingly (middle). Samples based on importance were selected and then advertised for classification(right).

avoids redundant transmissions that would repeatedly classify the same source (Figure 6.1). By minimizing the number of advertisements, the power demand on the transmitting side can be significantly reduced, as radio transmissions are vastly more energy-intensive compared to local computations performed by a microcontroller operating at a few MHz clock speeds.

One potential solution is to develop a technique that provides users with information about their daylong exposure while minimizing the amount of data that needs to be analyzed. In other words, the goal is to minimize the number of packets transmitted by the sensor node to conserve energy while ensuring accurate source classification throughout the day.

In this dissertation, we propose a novel method to minimize the offloading of raw sensor data for source classification by identifying events of interest and selecting important samples locally at the sensor's microcontroller. Specifically, we focus on detecting light-switching events from continuous RGB recordings—a task that is efficient yet challenging due to natural variations in light and interference under the same source.To address this, we develop a robust algorithm with carefully chosen parameters and thresholds to distinguish switch-over events from other realistic variations.

Successive RGB samples within timeframes are often comparable, allowing less critical samples to be reconstructed from neighboring values at the receiver. Following switch-over detection, the sensor identifies and transmits only the most critical samples for classifying the new source, balancing the trade-off between sample reduction and classification accuracy.

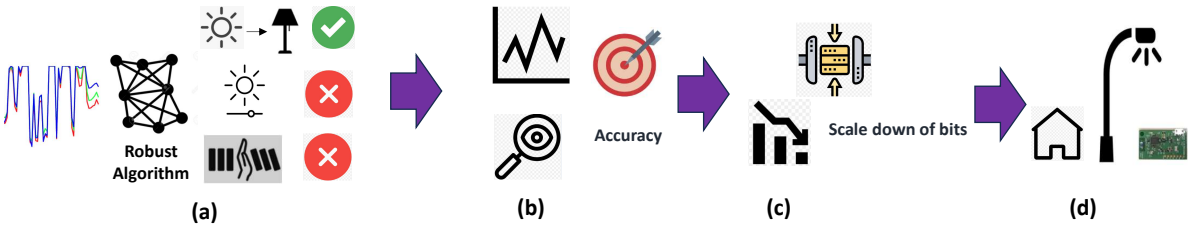


Figure 6.2: Data Efficient Roadmap: (a) Robust algorithm to separate light switchover events from long recordings and realistic variations under same source (intensity variation/ interference), (b) Detecting samples within timeframes that predominantly influence the classifier's performance, (c) Identifying appropriate and edge device friendly lossless compression technique and (d) Evaluating classifying performance pre and post procedures under realistic testbed

Furthermore, we propose a compression strategy based on sample value similarity to optimize information transmission for advertising purposes. Identifying an efficient compression method is essential for ensuring resource-efficient deployment on edge devices while maintaining data integrity. If not mentioned otherwise, the figures in the following sections represent recorded decimal value variations over time.

6.1 Relevant works with Limitations

Event-triggered architectures demand specific hardware for implementation that introduces unnecessary costs and design complexities. Furthermore, the threshold-based approach may prove ineffective in tasks such as identifying light sources, where fluctuations of parameters can arise from the same source or from adjusting the intensity of a source. Adaptive rate energy-saving data collecting techniques for WSN's have been proposed by Jaber et. al. [50]. The adaptive sampling approach has particular limitations where there is sampling rate/ maximum power limitations at sensor nodes. Autoregressive prediction (ARP) or Send-on-delta concept are inappropriate for light source data, which exhibits both static characteristics (like LED) and dynamic features (like Sunlight). Tarek et.al. have analyzed DCT and DWT based compression techniques where performances before and after data compression was measured in terms of PSNR, throughput, ETE delay and battery lifetime [51]. Although offering a high compression ratio, implementation requires considerable resources at nodes.

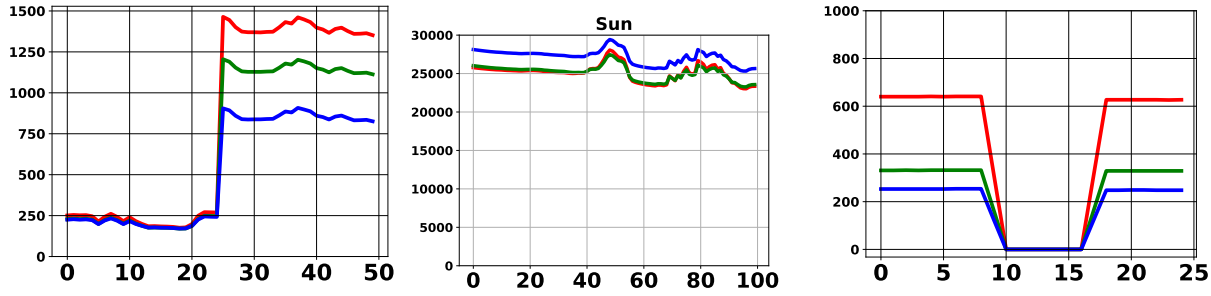


Figure 6.3: Real world RGB variations under same source: adjusted brightness of a LED source (left), recorded sunlight variation (middle) and random movement during data collection with CFL bulb on (right)

6.2 Considered Dataset for Analysis

For analysis, we pick our previous dataset collected by LPCSB. As described in chapter 3, the dataset exhibits significant diversity, incorporating data from both natural and artificial light sources. It includes 27 distinct artificial sources, along with natural light data collected under various conditions, such as different times of day, varying weather patterns, and a range of blinds and window glass types. For analysis, the dataset was employed to categorize indoor light sources into four distinct classes: *LED, Incandescent, CFL, and Sunlight.*

6.3 Experimental Design for Data Efficiency

The proposed system design consists of multiple stages focused on achieving optimal information compression. These stages include collecting and analyzing data from a primary sensor, detecting events of interest, performing selective sampling based on discriminative criteria, and ultimately employing lossless compression techniques.

6.3.1 Detecting Switch-Over of Indoor Lights

In real-world scenarios, surrounding lighting conditions can change without any actual switching of light sources, as illustrated in Figure 6.3. A robust event-detection algorithm must identify such incidents as variations within the same light source. To address this challenge, our objective is twofold: **(a) identify the parameter exhibiting the highest deviation during a true switchover event, and (b) establish a threshold for this parameter to distinguish genuine switchover**

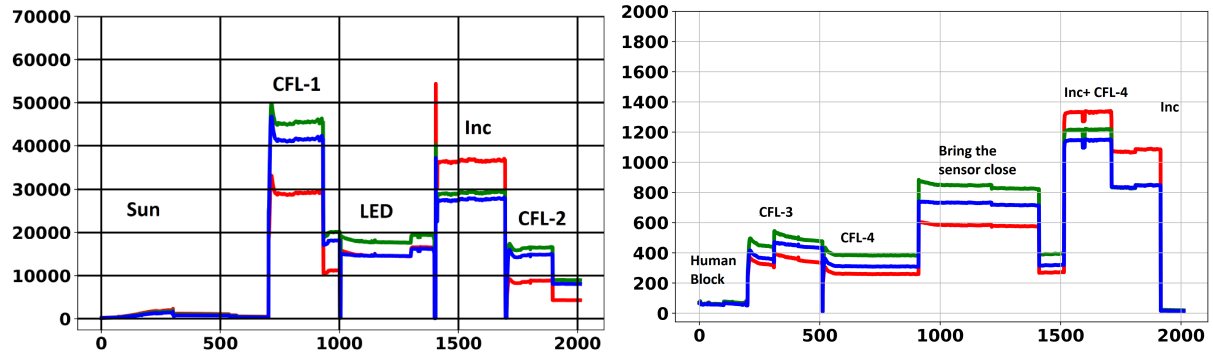


Figure 6.4: 2000 time samples from sensed RGB for two different types of lighting scenarios (Scenario-1: Indoor lighting at 5:00 pm, including single source/static env. with Sun, CFL and Incandescent bulbs (left), Scenario-2: Indoor lighting at 8:00 pm, multi source scenario with Incandescent and CFL bulbs and random movements (right)). Difference of *R/G*,*G/B* and *B/R* parameters between consecutive time samples were calculated. *R/G* was the most consistent and accurate.

events from the aforementioned variations. To achieve this, we intentionally alternate indoor light types during the data collection phase, modify the distance between the sensor and the light source, and introduce transient patterns by having participants walk past the source, thereby simulating real-world fluctuations in recorded values.

6.3.2 Best Parameter Selection

While adjusting distances or inducing transient patterns through sudden movements between the source and sensor, we observe changes in the magnitude of RGB parameters but no impact on their ratios. Hence, we opt to analyze the difference between consecutive ratios of fundamental color parameters ***(R/G, G/B, B/R)*** to differentiate switching from the mentioned events. Different sources have shown unique numbers and patterns in these ratios over time. When there is a switching incident of light, we expect to see a noticeable difference in those parameters before and after the switchover. This difference can be used as a tool to differentiate a switch-over event compared to other events.

To find the best indicator of switching in various scenarios, we set the indoor color sensor into single and multi-source scenarios (shown in Figure 6.4). The goal is to select the parameter that consistently exhibits the most variations across different scenarios. As we observe the difference of the ratios between successive time samples, we find that the ***G/B*** is less pronounced than the other two, whereas, ***B/R*** generates some false peaks, even when there has not been any change of light.

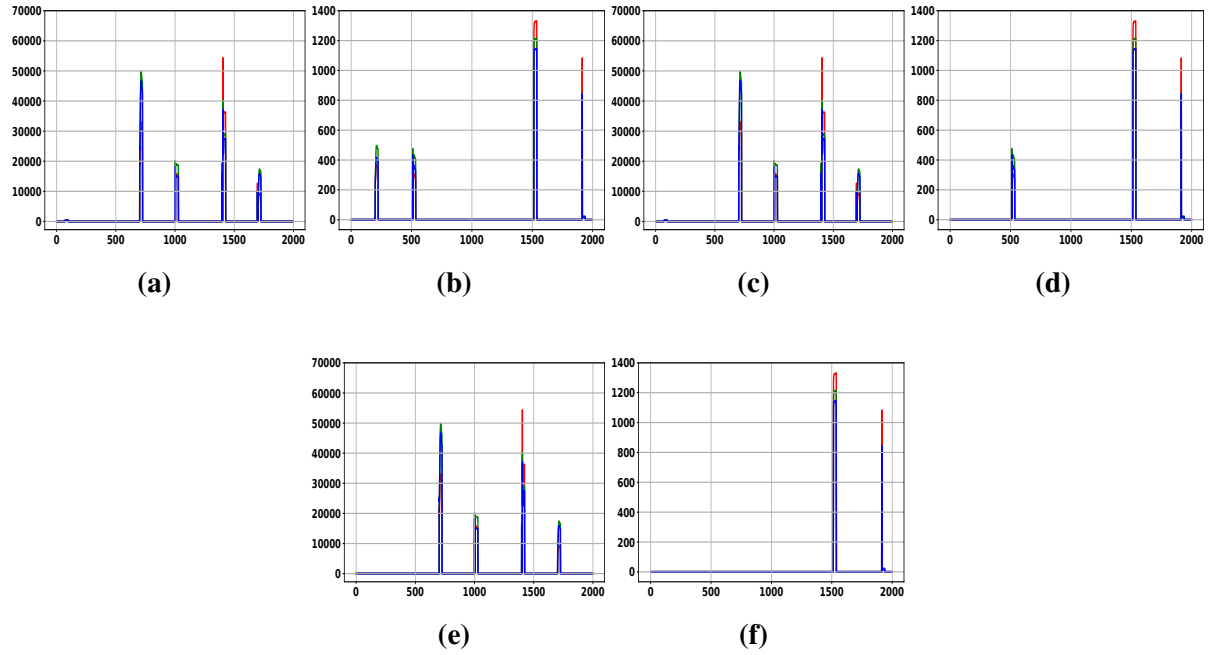


Figure 6.5: Varying the R/G difference threshold between successive samples for event detection for both testbed scenario: first order difference of *R/G* threshold was set: *1.5* for (a) & (b), *2.0* for (c) & (d), *2.5* for (e) & (f). As seen, setting threshold below *2.0* considers other variations, whereas setup at *2.5* misses events. So we proceed with first order threshold at *2.0*
So we continue with ***R/G***.

6.3.3 Threshold Setting

Variations in *R/G* can occur without switchover events, for example, during daylong variation of natural light or introducing a secondary source in the background. Therefore, we measure the difference of *R/G* between successive samples and try to determine a **threshold difference value** that can distinguish the switch-over events from incidents that generate noteworthy *R/G* variation. The precise selection of the threshold value is crucial. Setting the threshold too low may interpret color parameter variations caused by sudden changes in the background or natural fluctuations in the light source as events. On the other hand, a high threshold may result in missing switch-over events. Even after selecting the optimal first order difference threshold value for identifying the

switch-over events from incidents with similar variations, we observe that there is a challenge to separate the switch-over events from time specific natural light variation. Especially during sunrise

and sunset, sunlight can show variations in *R/G* across consecutive time samples that exceed the set threshold for first-order differences. For that, after setting the first order difference as the initial threshold, we observe **second order difference of the aforementioned parameter among samples, that is the difference between first order thresholds discussed above for consecutive points immediately after the detected event**. We note that while switch-over events show similar first-order threshold changes as natural light, the second-order thresholds over subsequent points are significantly more drastic, differing considerably from natural light variations. Therefore, we establish a second-order threshold to distinguish light switch-over from time-specific natural night variation. Any change in the mentioned parameter surpassing both thresholds is identified as a switch-over event, while other deviations are considered regular variations and are not considered for advertisement/classification. For first order threshold setup, we vary the threshold within the range **0.5-2.5**. The best result was observed with **2.0** (Figure 6.5).

As the first order threshold identifies *R/G variation* during sunset as switch-over incident, we observe the second order difference. For scene 1, we observe 4 peaks that satisfy the first order difference threshold within the timeframe. We then observe *R/G* differences immediate after the sunset peak and found values as ***[0.93078354, 0.92467979, 0.80124044]***, while for the rest of the peaks during light switching, first order *R/G* difference values were ***[1.20116618, 0.48860399, -1.14577259], [0.8567505, 36.25462304, 0.94773806], and [0.55755984, 20.15884116, 0.85261906]***. Similar numbers were also observed in scene 2. Therefore, we set second-order threshold for both first order differences at ***0.5***. That means, whenever the sensor will detect samples beyond first order threshold of **2.0**, then it look for whether they also satisfy second order

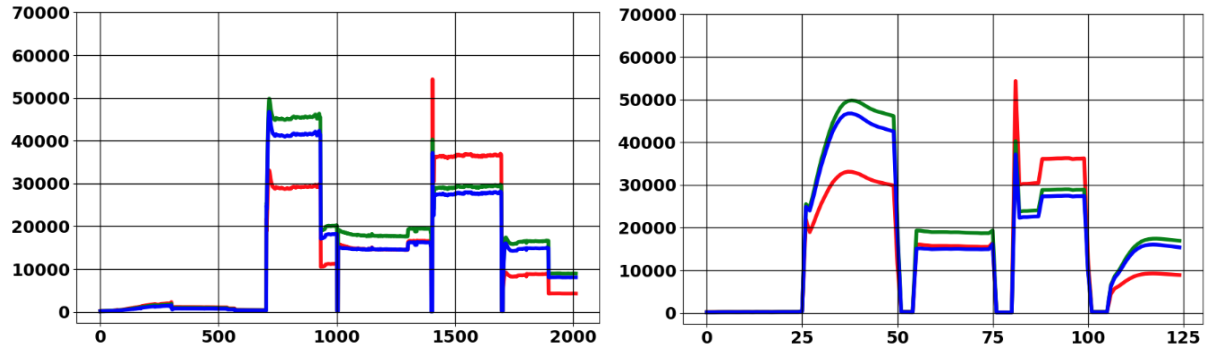


Figure 6.6: RGB samples before (left) and after (right) implementation of event detection algorithm

threshold of **0.5**. If both the thresholds are satisfied, it counts the incident as a switching event and considers 25 samples based on analysis described in chapter 3 to identify the new source. The modified timeframe after implementing this method for scene-1 is showed in Figure 6.6.

6.3.4 Particular Sample Selection

As **KNN** was reported as the top-performing classifier in terms of accuracy in chapter 3, we examine the influence of individual samples within timeframes on classification performance using the pre-trained **KNN** classifier. The procedure for figuring out the most important samples comprises of the following steps:

- Only a selected number of samples, chosen for their highest importance in classification will be sent to a remote device for KNN-based classification, omitting less crucial samples from the advertisement process to conserve energy on the device.
- At the recipient's end, we recommend using a pre-trained KNN classifier to enable off-board classification, thereby minimizing the on-device resource requirement for classification.
- Each advertised packet will contain the sample(s) number(s) within the timeframe, which will allow the receiver to identify which particular samples within the timeframe needs to be reconstructed.
- Following the reconstruction, the receiver will forward the reconstructed timeframe to the input of the pre-trained KNN classifier for classification.

For sifting the most important samples, we select test set randomly and implemented Permutation Feature Importance Method [52]. The process is described below:

- After training the KNN model, we pick a feature within a timeframe and across the test set to assess the importance.
- We then permute or shuffle the values of the selected feature across all test examples.
- The permuted dataset is then passed to the pre-trained KNN model. We then calculate the modified performance of the test set against the baseline test set accuracy before the shuffling.
- We rank the features based on the magnitude of the performance drop. Features that cause a larger drop in performance when shuffled are considered more important.
- Based on the required degree of compression, we pick n samples for advertisements.

After implementing the Permutation Feature Importance Method within the test set of the datasets, we organize sample points within timeframes based on their importance. As illustrated in Figure 6.7

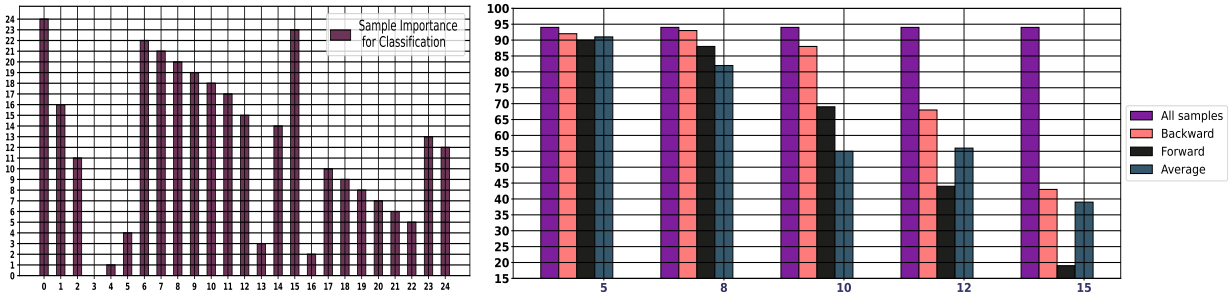


Figure 6.7: Importance of samples for classification (x-axis: priority of sample, y-axis= sample position in 25 sample timeframe). Lower index on x-axis means higher importance in classification. Recalculated identification accuracy from post signal reconstruction with KNN classifier, compared against baseline test set accuracy (purple bar) (x-axis: number dropped and then reconstructed samples out of 25, y-axis= accuracy in percentage (right))

(left), our analysis indicates that **there is no direct correlation between a sample's position in a given timeframe and its importance level for classification.** Subsequently, we investigate the impact on KNN accuracy when leaving out a specific number of samples and reconstructing them at the receiver, as illustrated in Figure 6.7 (right). We observe that **the number of missing samples is not always linearly proportional to accuracy**.

6.3.5 Lossless Data Compression

Following the exclusion of a specific number of samples, our aim is to further compress information, transform it into lower number of bytes and accommodate them in fewer packets to minimize the total number of advertisements. For that, we decide to investigate lossless data compression methods for information minimization. By considering ease of implementation on edge platform and the nature of our dataset, we investigate ***Huffman, Run-Length Encoding and Arithmetic Coding*** algorithms. As grouping samples with the same values may compress the information further, we implemented compression both with and without grouping samples, and the results are compared against the raw data (Figure 6.8). We record the mean number of bytes considering inter/intra-class variation and for both regular and transient patterns for comparison. As seen in (Figure 6.8), **Huffman** technique excels among three methods. Later, we execute best performing **Huffman** and **Arithmetic** method on edge device for comparing the performance. As seen, although **Huffman** method requires additional time for execution, it requires fewer resources for on-device deployment than **Arithmetic** method. We pick **Huffman** method for analysis.

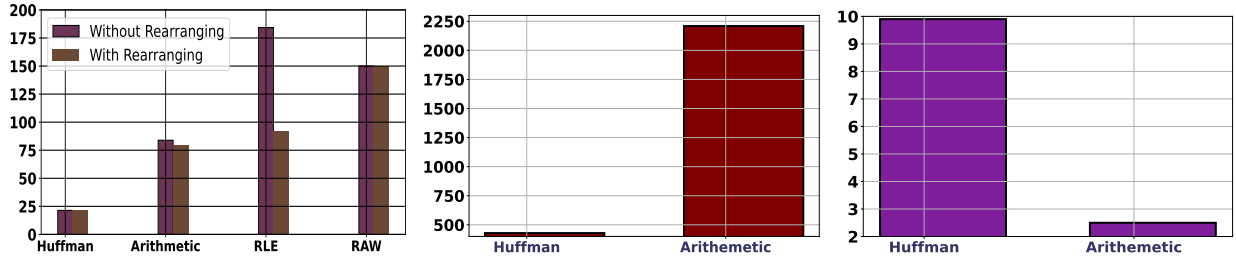


Figure 6.8: Comparison of information reduction for different lossless reduction methods (RGB: left) (y-axis: byte count). Results depict that Huffman coding was the best performer. After execution, memory utilization (middle)(y-axis:KiB) and execution time (right) (y-axis:in microsec) is also demonstrated for Huffman and Arithmetic methods

6.3.6 Reconstruction

Since our classifier is pre-trained with 25 samples, we must reconstruct the discarded samples, which were deemed less important for classification, on the receiver side. We opt for three simple reconstruction techniques: ***Forward Filling, Backward Filling, Average Filling***. We analyze trade off between number of samples considered and classification accuracy after reconstruction for each of the methods. Based on the analysis in Figure 6.7, **Backward Filling** was the top performer among three for the same number of missing samples. A comparison of pre and post-reconstruction of timeframes reveals that while the general shape of the signal can be reconstructed, **the post-reconstructed signals have lost a certain degree of smoothness compared to the original timeframe (Figure 6.9)**.

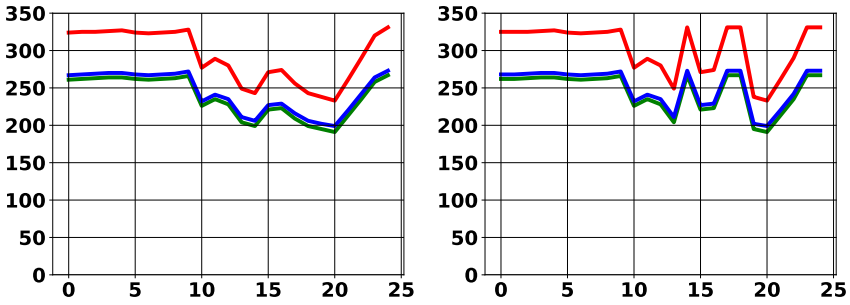


Figure 6.9: Comparison of timeframes pre and post-reconstruction: The initial RGB signal (top left) underwent reconstruction, addressing 8 missing points out of 25 using Backward Filling

Algorithm 1 Switch-overs for Source Classification

Start:

-Select R, G & B values from the sensed values

-$selective = t_{a_i}$\{Discriminative RGB out of 25\}

-$set, n = 0$

- Calculate $(R/G)_n$ \{R& G ratio of nth sample\}

- x, y: First/Second Order threshold

if $n \leq 25$ **then**

  -$set, i = 0$ \{Initial Lighting\}

  for $i = 0$ to 25:

  if *n+i= selective[i]* **then**

    Save sample

  end

  Encode and Advertise

  else

    Do not advertise

  end

end

else if $n \geq 25$ **then**

  $count = n$

  $th_n = |(R/G)_n - (R/G)_{n-1}|$

  $th_{n+1} = |(R/G)_{n+1} - (R/G)_n|$

  $th_{n+2} = |(R/G)_{n+2} - (R/G)_{n+1}|$

  if *($th_n \geq x$)* **then**

    Calculate, $diff_1 = |(th_{n+1} - th_n)|$

    Calculate, $diff_2 = |(th_{n+2} - th_{n+1})|$

    if *($(diff_1 \geq y)\&(diff_2 \geq y)$)* **then**

      -$set, i = 0$

      for $i = 0$ to 25:

      if *(n-count+i)= selective[i]* **then**

        Save sample

      end

      Encode and Advertise

      else

        Do not advertise

      end

    end

    else

      Do not advertise

    end

  end

  $count = count + 25$

end

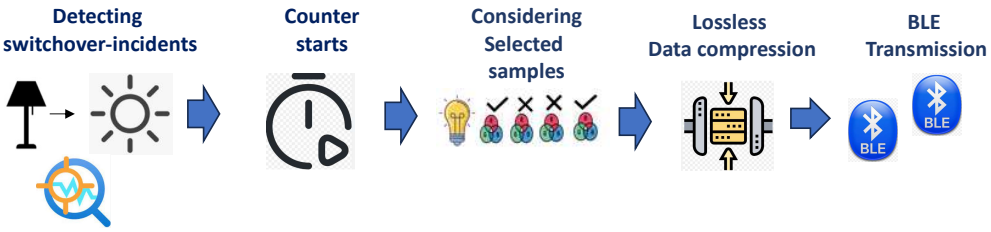


Figure 6.10: Proposed tasks to be performed on device

6.4 Proposed Implementation Architecture

For resource-efficient operation, we propose implementation of smart switching detection, sample selection and encoding part to be done on-device, as depicted in Figure 6.10. Light switching detection algorithm is described in **Algorithm 1**. After activation, proposed smart indoor light sensor will collect pre-defined number of samples (here, that is 25), select samples out of them based on their positioning, decode and advertise with minimum advertisement packets for the initial lighting environment detection. For extended periods of data collection, the sensor will continue to cease advertisement until the specific criteria for switching has been met. Upon meeting the criteria, the sensor will reset the counter to the initial position of detected deviation due to switching, choose particular samples from the next set of 25, encode them, and subsequently advertise them. Encoded packet is proposed to contain sensed RGB information with packet numbers.

Receivers will initially receive the advertised samples, decode the encoded signal, assess which signals are missing from timeframes and necessitate reconstruction, and finally, execute classification (shown in Figure 6.11). It's assumed that the receiver side has sufficient resources for reconstruction, classification, and storage/display of results as needed.

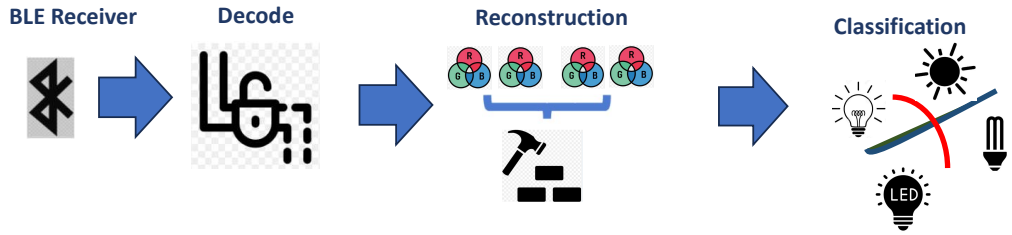


Figure 6.11: Proposed tasks to be performed off device

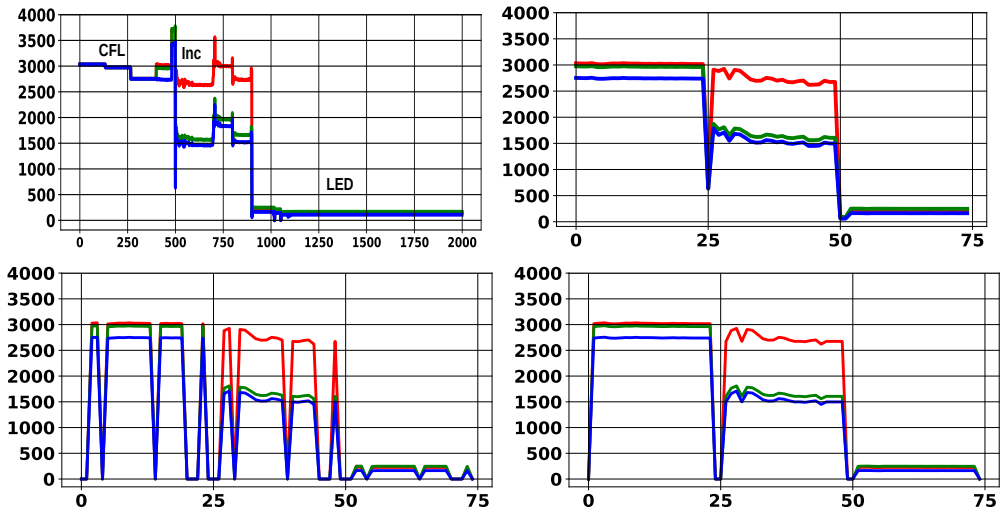


Figure 6.12: Implementing Light sensors in a real-world setting: (top-left) collecting continuous RGB data from indoor environment containing multi-type sources, (top-right) separated segments for source identification using the event detection algorithm, (bottom-left) Signal focusing on only most important 17 samples out of 25, (bottom-right) Reconstructed signal utilizing the Backward Filling algorithm for event classification

6.5 Evaluation

Ultimately, we deploy indoor light sensor in entirely novel testbed setting with the exact experimental setup discussed in chapter 3. The RGB testbed encompasses indoor scenarios with three types of artificial lighting featuring random switching and movement the source and the sensor (Figure 6.12). Throughout the data acquisition phase, the total raw information is quantified in bytes (Color sensor: 6 bytes per RGB sample). Next, we carry out event detection, selectively choose 17 samples out of 25 from the identified event section to maintain performance threshold above 90%, apply Huffman compression and ultimately, calculate the number of bytes before and after implementation of the process. After applying the Backward filling algorithm (*for RGB*) to reconstruct the signals, we then use these reconstructed timeframes as inputs for the pre-trained KNN classifier. Notably, the classifier correctly classifies all the indoor light types, even with significant compression of the raw information (Figure 6.13).

Full Length Raw RGB(bytes)	*Full Length Lossless Comp. (bytes)*	*After Event (bytes)*	*Samp. Select*	*After Select (bytes)*	*Lossless Comp. (bytes)*	*Data Eff. Raw*	*Data Eff. Comp.*
12000	6321	450	17 (out of 25)	306	247	***97.94%***	***96.09%***

Figure 6.13: Evaluating the data efficiency (in percentages) from raw information to the final compressed version ready for advertisement. Upon decoding, reconstruction, and classification process, all the three artificial bulbs were accurately classified

6.6 Comparing Reported Data Efficiency with Previous Works

In the following table, we present a comparison that how much data efficiency we have achieved with the **Dimension Reduction** and the **Daylong Data Efficient** techniques. Table 6.1 compares several previous Light Source Classification studies for reference. As shown, most of these studies incorporated additional sensor data along with RGB sensor data. This includes spectral data consisting of a large number of channels (wavelengths) that require substantial memory space and computational power. So a single benchmark for reporting overhead information is absent here. Although exhibiting poor accuracy, we can assume Sarris et.al. work as our baseline, as the author has utilized only RGB data for classification. Here, each color value from RGB is represented using 2 bytes. If we utilize only 25 RGB examples for source classification, it would require processing of 150 bytes of information.

To evaluate data efficiency, we use two metrics: (a) the ratio of bytes between the full-length raw RGB data and the final compressed bytes required for advertisement, and (b) the ratio of bytes if lossless compression **(here, Huffman)** is applied to the full-length signal, using the same denominator as in (a). As seen from Table 6.1, the data efficiency reaching **up to 97.94%** and **up to 96.09%** with for metric (a) and (b) respectively.

6.7 Energy Efficiency with reduced data

To give an idea that how data efficiency impacts power intake, we calculate the current draw using the tool **Online Power Profiler for nrf51822 Bluetooth LE** for different payloads in the

advertising mode. As seen, the average current draw also reduced around **89.71% compared to full length observations** (Figure 8.21). This significantly increases the operational duration of the indoor light sensor under energy budget.

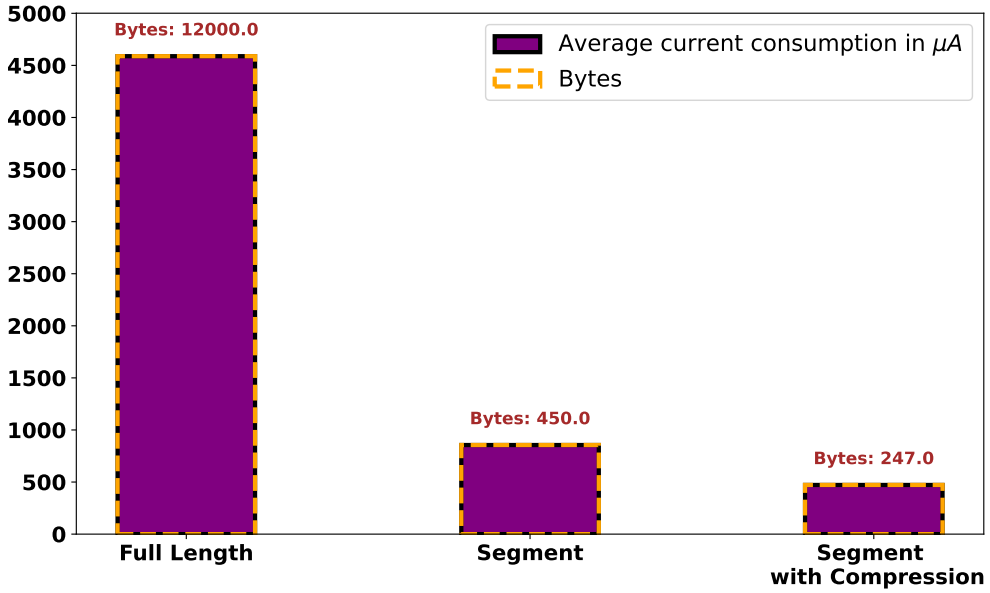


Figure 6.14: Comparison of average current draw in μA for different payloads. As seen, after switching detection, selective sample consideration and lossless compression, energy draw was reduced significantly

In our work, we utilize the same parameters, but with the abovementioned approaches, we were able to elevate the classification performance in both ideal and realistic scenarios.

Systems	**Source Types**	**Sensors**	**Parameters Studied**	**Chosen Parameters**	**Reported Highest Accuracy**	**Bytes Required for/ Overhead Considered**
Indoor Light Conditions [31]	Halogen, Fluorescent, Warm LED, Cold LED	TSL2561, ISL29125, AM1815CA, POW111D2P, USB2000+	Luminosity, RGB, Solar Panel, Spectrometer	Spectrum [(300-1100)nm (620-1080)nm], R,G,B	100% (Ideal Condition) 84.21% multi-source scenario 62.5% (outside training range) Classifier: Subspace Discriminant Classifier	Spectral+Raw (Lux, RGB, Solar) No
Importance [53]	LED, Halogen, Sunlight, Fluorescent	AS7262, TSL256, BMP180, ISL29125	SIX-color channels (GIBOVR), Intensity, Temperature, RGB	All	100% (Ideal Condition: Direct), 90.2% (Ideal Condition: Indirect), 47.2% (Indirect: Removed Violet Channel) Classifier: Cubic SVM	Raw (Lux, Color, Temp) No
Sarris [39]	Incandescent, LED, Sunlight, Fluorescent	TCS34725	RGB	RGB	Highest: 96.1% (Sunny day) Lowest:76.4% (Cloudy, evening) Classifier: Threshold-based	Raw RGB No
PV Harvesting [32]	LED, Fluorescent	UTSL2561, ISL29125	Spectrum [Broadband, Near Infrared, Narrow Near Infrared], RGB	All	96.8% (Ideal), 50% (switching) Classifier: KNN	Spectral+Raw (RGB) No
Our Method	**Inc, LED, Sunlight, Fluorescent**	TCS34725	RGB	RGB	99.2% (Known) 90.8% (Switching), 90.2% (realistic testbed including switching & unseen Sources) Classifier: KNN	Raw RGB **Yes** Dimension Reduction (97.33% Raw RGM comp), Switching, Sample Selection, Lossless Compression (97.94% Raw RGM comp)

Table 6.1: Comparison of Light Source Classification Relevant works with Our Approach: Introducing Overhead Information Reduction

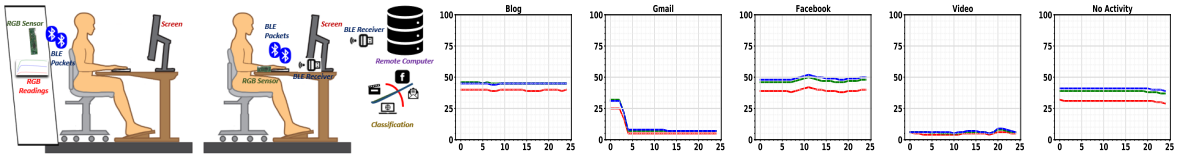
Chapter 7

SCREENSENSE: ONSCREEN PASSIVE SENSING FOR USER IDENTIFICATION

In recent years, researchers have related escalated engagement on the screen to cognitive and behavioral disorders, depression and anxiety, obesity, physical issues like Computer Vision Syndromes (CVS), neurodegeneration, circadian rhythm, and attention disorder [54]–[58]. Collection of screen activity data over a prolonged period for behavioral and physiological analysis possesses a variety of methodological, security, and privacy challenges. It includes storing memory intensive high resolution data or breaching confidentiality through distant screen recording [59].

Gathering privacy intact screen usage is manageable through users' feedback, which has been found inaccurate more often than not. Utilization of customized software/platforms (*such as third-party data sources, cookie-matching technologies etc.*) for anonymous enlisting is another option. However, such method can expose extraneous browsing details, and raise skepticism regarding openness of other confidential information inside a device.

Researchers have also shown that characterizing online activity can be accomplished from a distance and with passive sensing, like from keystrokes. *So why we want to use indoor light sensors for screen activity analysis?* There are multiple reasons for that. First, passive sensing with majority of these methods require additional setups or are generally used in a manner that can cause discomfort for continuous, daylong operation. Additionally, they must be synchronized to switch on and off in alignment with the monitor's activity status. The use of general-purpose indoor light sensors eliminates the need for additional task-specific screen sensors that introduces additional energy consumption, or the arrangement of customized software settings for screen activity classification. Instead it allows individuals to leverage the **existing framework** for sensing. The



(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Figure 7.1: (a) Proximity Sensing of on-screen activity through *BLE based Color Sensor*: Sensor can be installed on a wall just behind the user or be carried as a wristband device. Sensed parameters can then be advertised through *BLE packets* and offloaded to any nearby or remote computer for *Processing and Classification*. Variation of sensed RGB patterns for (b) Blog titled *"Introduction to Attention Mechanism"* by *Kemal Ardem*, (c) Gmail under *"Dark Black Theme"*, (d) Facebook with *"Dark Mode off" setting*, (d) Movie sequence from *"Bourne Ultimatum"*, (e) Screensaver mode with *"bubbles"*. *x-axis*: number of samples, *y-axis*: recorded RGB values

same device can also be utilized for analyzing light source exposure throughout the day. This approach aligns with the Green IoT concept by enabling the use of a single device for multiple applications. Moreover, the capacity of these indoor light sensors have not been thoroughly investigated. Specifically, when positioned near a user's screen, it remains unclear whether they can reveal on-screen activities, potentially compromising the user's privacy.

Before considering Indoor Light Sensor for sensing, we consider the feasibility by addressing two major questions: *Is it possible to utilize basic color information from a general light sensor for the classification of screen activities?* And with the provided information, *how can we effectively navigate real-world factors that might jeopardize the integrity of this approach.?*

To address these questions, we propose `ScreenSense`, an innovative platform designed to detect user activities through passive sensing. In the `ScreenSense` project, we deploy LPCSB in proximity to the computer screens for passive on-screen activity sensing. To conserve onboard resources, particularly for classification tasks, LPCSB was again set to transmit collected sensor data to a remote platform, such as a nearby cloud server, for further analysis. Screen usage data from selected categories were gathered by placing the board in controlled indoor environments and near screens across various settings. To reduce calculation complexity, rapid processing, and avoid unnecessary data storage, `ScreenSense` simply utilizes raw variables from sensors for classification.

Besides enabling passive activity monitoring, `ScreenSense` raises a **privacy implication of indoor light sensors** which are mostly installed naively without considering the possibility of

leaking privacy-sensitive information. This opens the door to further research regarding how to minimize color data side-channel effects during design, deployment, and data communication process. Below we summarize the contribution of `ScreenSense` project:

- We propose `ScreenSense`, which utilizes **existing architecture for light sensing** to detect and categorize users' on-screen activities into five distinct classes:***Mail, Social, Reading, Video and No activity***. For activity classification, we leverage **only raw information** advertised through BLE packets after placing the device in front of the screen.
- We consider **several practical factors** including activity data from different users, various types of screens, screen settings, operational settings, and nearby light source effect to design a detailed and realistic screen activity database. To classify captured **non-linear RGB information from screens**, we study several machine learning-based approaches including neural networks, and time series-based architectures with fine tuning to find the best performing classifier model with **a minimum number of data packets.**
- We implement several **data augmentation techniques** to address real-world performance limiting factors. Our experiment demonstrates a significant performance improvement across all testbed scenarios **up to 7.5%** after performing the augmentation.

7.1 Related Work and Limitations

In this section, we discuss related approaches to monitor users' screens activity. We categorize major screen monitoring techniques as follows:

7.1.1 Self-reporting and Software based approaches

Self reporting activities like [60] were largely found to be inaccurate and confusing, which questions the credibility of this approach [61]. Utilization of customized software/platforms such as third-party data sources and cookie-matching technologies have also been adopted because of more precise and accurate tracking [62], [63]. Unfortunately, this requires additional installation for every device a specific user comes across during the day, potentially disclosing superfluous browsing information and even posing the security risk for the classified information within a device if not implemented carefully [64].

7.1.2 Indirect Sensing Mechanisms

Researchers have also shown that characterising online activity can be accomplished with *specially designed eyeglass [65], head-mounted color light sensor [33] and Keyboard extraction,* [66]. Such indirect methods have multiple disadvantages. For sensing, these devices require additional setups or must be used in a manner that can cause discomfort for continuous, daylong operation. Additionally, they need to be synchronously switched on and off with the monitor's activity status. In addition, the eye level sensor is not quintessential for computer activity usage detection.

7.1.3 Screen Recording and Snapshots

Screen activity recognition through screen recording and taking snapshots have been attempted in [67], [68]. However, storing high-resolution snapshots of the screen over a prolonged period is memory intensive and privacy compromising. [59], [66]. They also demand complex framework design and post processing, which makes them unsuitable for mass deployment.

7.1.4 Choice of parameters for classification

Classifying activities based on lux information was attempted in [69]. As lux intensity can vary based on screen size, placement of sensors, and sensor configuration, classification based on it is going to be incorrect. Within the same setup, we discover that different activities share common RGB spectra and magnitudes, which makes it challenging to differentiate solely based on RGB thresholds as shown in Figure 7.2. t-SNE visualization of RGB values also reveals that dissimilar light sources are linearly inseparable. This necessitates careful classifier architecture selection like Machine Learning, Neural Network or Time Series Classifiers based algorithms, which are efficient in classifying non-linear time-varying signals.

7.1.5 Memory/Power Inefficiency

Spectra from mini spectrometers can reveal specific activities running on the screen [70]. However, such devices are expensive, power-hungry through unnecessary components like accelerometer, and unsuitable for mass deployment. For screen activity sensing to be practical and ubiquitously adopted, the sensor should be cost-effective, small, and standalone. For IoT devices as sensors, they are expected to optimize available resources to facilitate energy harvesting. For long-term recording and memory friendliness, classification should be based on minimum parameters.

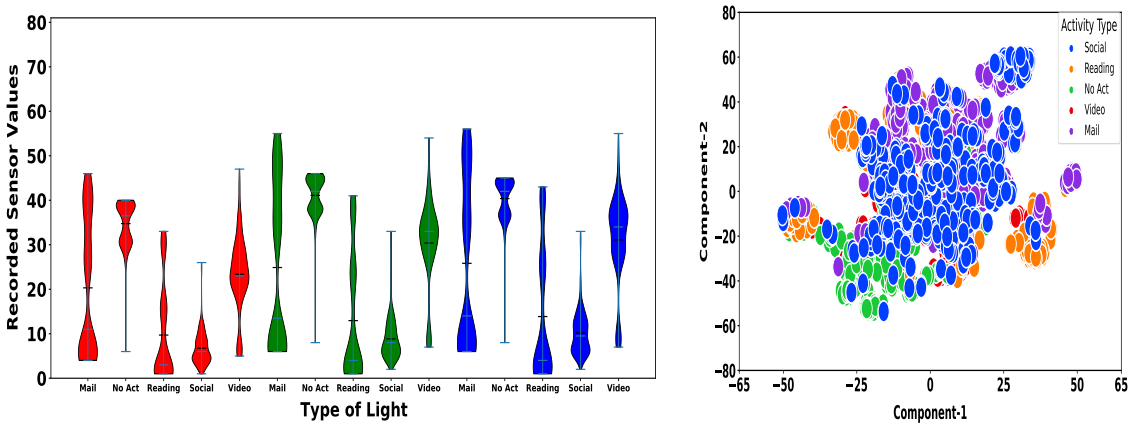


Figure 7.2: RGB violin plot of activity data (top) (x-axis:source type, y-axis: recorded values). Separate activities generate common readings and cannot be segregated solely based on threshold values. 2D tSNE plot with first two principal components exhibit linear inseparability (bottom)

7.2 Implementation of `ScreenSense` System Architecture

7.2.1 System Overview

`ScreenSense` design consists of placing LPCSB near a device use's screen and an activity classification framework that aims to design a high performing real-world activity detection platform. Figure 7.3 illustrates the basic blocks of operations. Our design consists of two major parts: dataset collection and dataset augmentation. After sensing, `ScreenSense` is designed to transfer the sensed parameters to a distant platform to minimize on-board resource requirements *(like, cloud)*. Major resource-intensive tasks, like pre-training of the classifier with the diversified dataset, along with identifying recent observations is performed at the distant device. The upper block of Figure 7.3 depicts the scope of data diversity incorporated within the training set, while the lower section illustrates the approaches used to acquaint that classifier with the real-world variations of the collected diverse dataset. Table 8.3 compares our proposed method against similar approaches. `ScreenSense` excels in several areas than similar approaches: it effectively addresses relevant screen activity categories, offers energy efficient passive operation with a wider range and employs significantly fewer parameters for data-efficient classification, thereby minimizing computational complexity and avoiding unnecessary delays. Data collection process

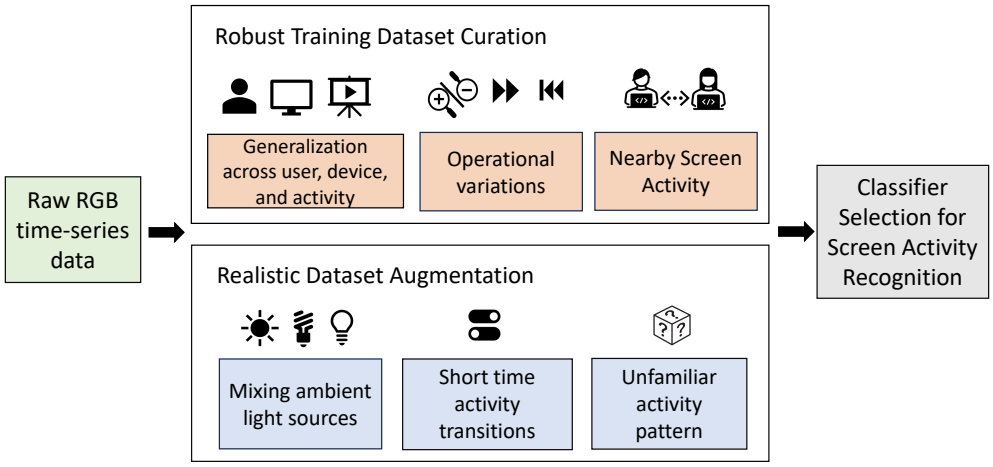


Figure 7.3: An overview of the `ScreenSense` framework

with the sensor in realistic indoor environments have been discussed in the *Experimental Methodology* section. Unless otherwise specified, the plots exhibit **recorded decimal values *(y-axis)* against time samples *(x-axis)***.

7.3 Implementation

7.3.1 Data Collection: Operational Range with Gain Settings

To collect the BLE advertisements, we use a BLE receiver **bled112** connected to a nearby computer. This computer also processes sensed parameters and runs the classifier. LPCSB transmits BLE data packets containing an ID and raw data including three different color filters (red, green, and blue) and no filter clear component. Recorded decimal numbers are proportional to the intensity of each component. Before collecting the data, we configure the advertising rate of *BLE packets* for Apollo.

Regarding sensor placement, our goal was to verify multiple issues: the operational range (distances/angles) of typical indoor light sensors, whether placement variation causes dissimilar patterns or not, and finally, for best output, in what fashion the sensors should be installed/used. From our observations, we determined that with custom screen settings and with only 1X gain setting, **LPCSB was able to collect data up to a distance of 1 meter from the screen.**

To mimic realistic scenarios and based on [71], [72], **we placed the sensor at a distance ranging from 10 cm to 100 cm and at an angular variation from 0-90 degrees relative to the screen**

Systems	Activity-specific detection	Installation requirement	Sensor requirement	Range	Power limitation considered	Parameters for classification
Tiger [65]	Screen View vs Non-view	User-worn (eyeglass)	RGB, IMU, Lidar	80 cm	No	RGB, Hue, Saturation, Intensity
LuxLeak [69]	Ten popular websites	User-worn (eyeglass)	RGB light sensor	60 cm	No	Lux
WISEGlass [71]	Watching Movie/Browsing	User-worn (smartwatch)	Light intensity	30 cm	No	13 statistical features from 23 virtual channels
Head Mount [33]	Document Reading	User-worn (forehead-mounted)	RGB light sensor	cm-scale	No	Same as above
ScreenSense	**Activity class**	**User-worn/Fixed deploy**	**RGB light sensor**	**1m**	**Yes**	**Only RGB and Brightness**

Table 7.1: A comparison of `ScreenSense` with the most relevant approaches

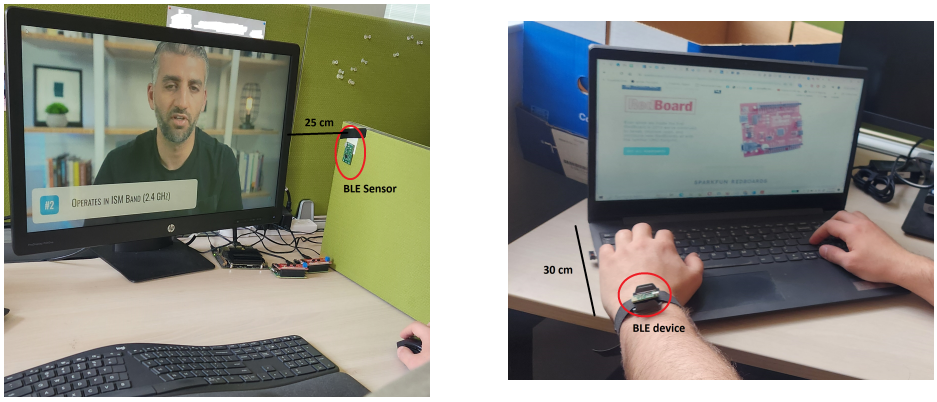


Figure 7.4: `ScreenSense` using LPCSB (top), Deploying the LPCSB in a personal workspace as a fixed-point installation (bottom-left) and as a handheld device (bottom-right) during data collection

(both screen tilted and wrist tilted). As seen,the sensor below the distance of 15 cm and the angle of 20 degrees records a too low magnitude to analyze with a 1X gain setting to analyze [Figure 7.4]. Higher gain settings can be utilized to gather information in low-light conditions or to extend the sensing range.

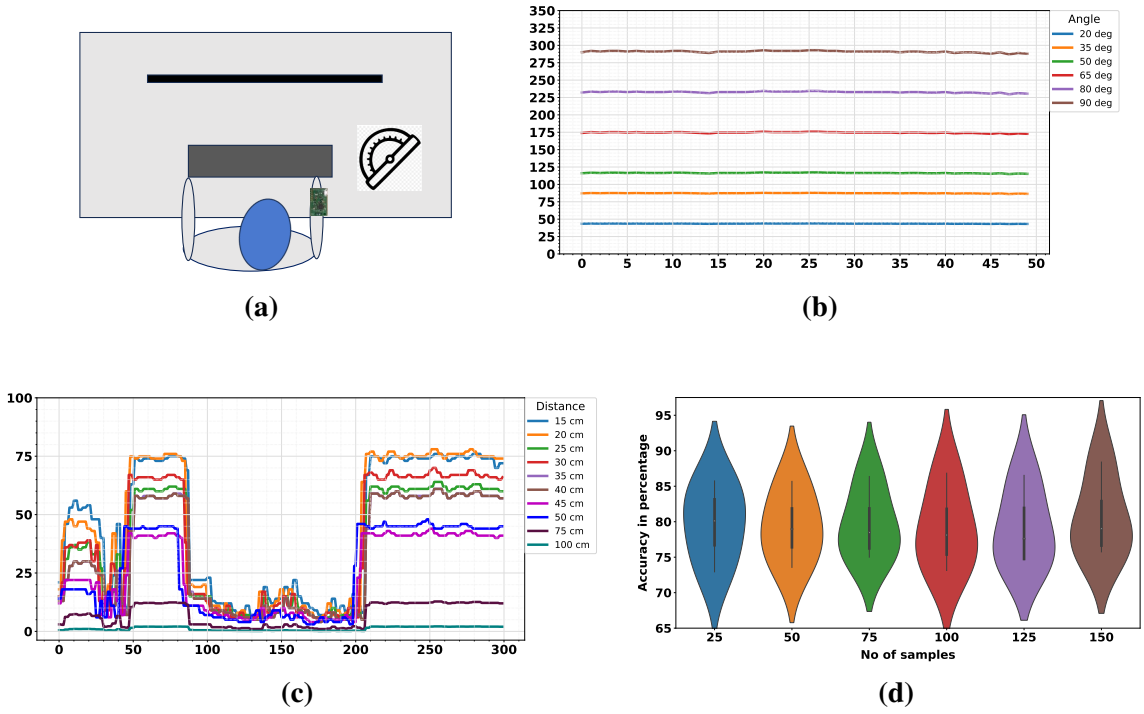


Figure 7.5: (a) Analysing angular and distance variation in wearable fashion, Recorded *Blue components* while playing the same video and placing the sensor at different angles (b) and distances (c). Pointing the hand horizontally and placing the hand at a distance of *20 cm* from screen records the largest values. As observed, both the angular and distance variation record similar RGB variations with different amplitudes. With 1X gain, sensor records RGB variations upto 1m . Accuracy with variable length window is shown with violin plots (d). Higher mean accuracy was recorded with 25 and 150 samples.

7.3.2 Choice of Sampling Rate, Ideal Sensor Deployment and Number of Samples for Classification

At first, we set the advertising rate of *BLE packets*. Settlement is crucial as the speed of operation and the frequency of switching among activities can vary from one person to another or with the same person at separate times. At the same time, setting an unnecessarily higher advertising rate can drain out the power supply much faster than expected. However, considering statistics on average stopovers on a website [73], the persistence of human vision and recording adequate fluctuations of RGB values for activity identification, we set our device to advertise 5

samples/second after sensing.

In this study, we capture screen data from various distances in different settings. We also performed experimentation for sensor placement at different distances for the same setting. Our goal was to verify two issues: (1) whether placement at various distances records dissimilar patterns or not, as it creates different sensing angles, and (2) when the user has the freedom for placement, where he/she should set the sensor to record variations with the highest magnitudes. The activity data on the screen can record a very small number, where better placement ensures that the minute fluctuations are captured, which can improve the classification performance. In addition, we observe that *variation in sensor placement generates similar patterns* and *the recorded values are not inversely proportional to distance* (Figure 7.5).

Our next step was to determine the optimal number of samples for classification. Too few samples may prove inadequate for classification, while too many samples can be memory intensive and will needlessly delay the classification process. We start by randomly picking up a fixed number of observations from the training set. Next, we divide the original time series observation into multiple equal-length sub-sequences by choosing six different length slicing windows (consisting of *25,50,75,100,125,150* RGB samples). With overlapping and non-overlapping samples, all those windows contain an equal number of examples. We apply four different ML algorithms for classification, Decision Tree (DT), Random Forest (RF), K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) with Radial basis. The best performance was observed with window of 150 samples. However, accuracy with window of 25 samples was the second best, which offers classification with significant less time and power requirements. Proceedings were then continued 25 samples (Figure 7.5). As our goal was to enable classification with minimal information, we start our analysis only with RGB values. However, as discussed later, after analyzing misidentified examples, we later add clear components of the observation for betterment.

7.3.3 Dataset Overview with Labeling

At first, we try to utilize the existing database of screen sensing. Unfortunately, all the RGB based databases were either purpose specific (e.g., TV on off with distance in [74] or with limited categories/ inappropriate with our purpose (e.g., 800 images from five educational categories in [67], [75]). For that, we decide to collect our dataset and select the top four screen activity classes: **Mail, Video, Social, Reading**, along with inactive hours **class: No Activity** based on screen usage

report **discreen**. We argue that detecting these classes of activities instead of identifying specific applications or websites (Facebook or Gmail) is more insightful for many applications including personal productivity tracking, work-life balance, and attention span of users.

Labeling datasets can be confusing for multiple reasons. There is a wide diversity in screen activities, making it challenging to draw clear boundaries (for example, determining whether ”watching a video on Facebook” should be categorized as Video or Social?). In addition, multiple activities occur simultaneously. To simplify matters, we assign an activity to its primary platform (like playing a video on Facebook is labeled as ”Social”) and displays only one activity on one screen at any given time (e.g. [76]).

7.4 Composing Training Dataset

In training dataset generation, we identify inherent variations in the RGB recordings even in dark room environments related to the screen, screen setting, operating nature etc. that can impact any passive screen activity identification approach and need to be addressed for better classification. Including these variations in the dataset aims to familiarize the classifier with realistic scenarios as comprehensively as possible. However, it is not feasible to encompass all these variations in the dataset. Therefore, our approach emphasizes analysing and incorporating RGB variations associated with selected events into the training set to enhance robustness, marking a novel exploration in this area.

To generate inter and intra-class variability of on screen activity data, we have considered the following variation in our training set:

-Screen to screen variation
-Screen settings variation
-Same class variation

7.4.1 *Screen to screen variation*

Dimensions and features of different screen can result in variation of sensed values for the same activity. As seen from figure 7.6, the same activity sensed at different screens differs from each other.

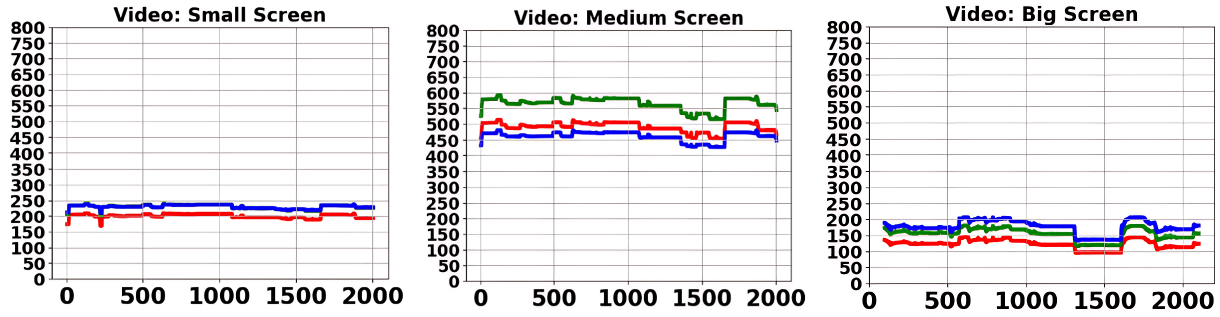


Figure 7.6: Same video sequence played on *Lenovo Ideapad S145 laptop with 15" monitor* (left), *Dell 32" computer monitor (middle)* and *LG 55" display (right)*. x-axis: sample no. y-axis: recorded values

7.4.2 *Screen settings variation*

People can use different screen settings (*brightness, theme* etc.) based on their necessity and preferences, which results in variability of sensed values. For that, we collect data by setting screen with different settings, as seen in Figure 7.7.

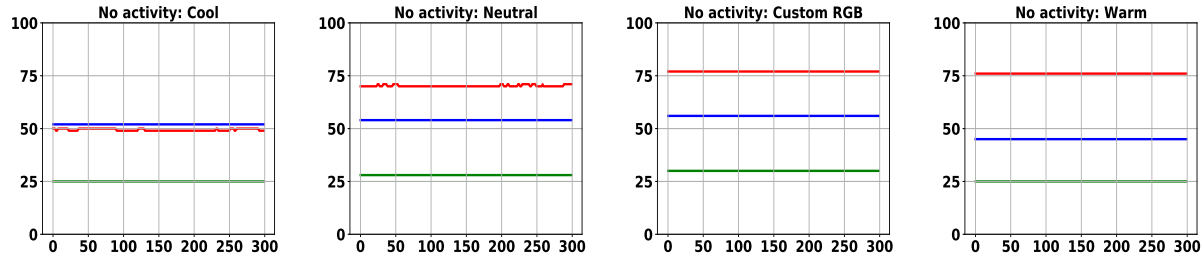


Figure 7.7: No activity RGB variation of 300 samples, collected from a *HP 32" monitor* while the display was set on four different color modes: *Cool, Neutral, Custom-RGB and Warm*

7.4.3 *Same class variation variation*

Activity from same class can have multiple variations. For example, social interaction can be performed using various platforms *Facebook, LinkedIn* etc., which we cluster as social class. This goes same for other classes (Mail: *Gmail, Outlook, Yahoo Mail etc.*, Video: *Movie, Gaming, News, Songs etc.*) and so on. Even with screensaver mode, different settings like random flyers, bubbles etc., can generate contrasting patterns. We have captured the aforementioned variations in our training set as intra-class variations (Figure 7.8).

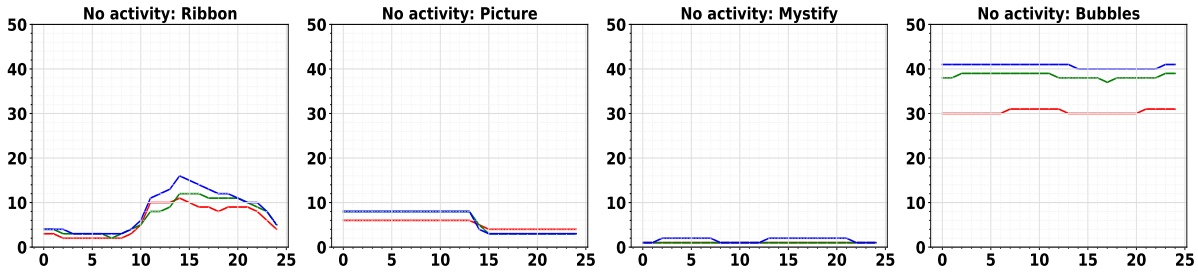


Figure 7.8: Analysing RGB variation of different screensaver setup: (a) Ribbons, (b) Pictures (c) Mystify (d) Bubbles. x-axis: sample no. y-axis: recorded values

7.5 Real World Variations

In real world, operating style of an activity *(for example, playing a video at double speed)* or the screen surroundings can influence the recorded RGB patterns. In this section, our goal is to analyze a few of these variations and study the fact that if needed, whether it is possible to regenerate them using libraries.

For analysis, we consider four types of variations. The first three variations are generated by the user and the last variation is initiated by the surroundings. Man-made variations are chosen to be action specific *like regulating video playback speed or zooming the screen while reading*.

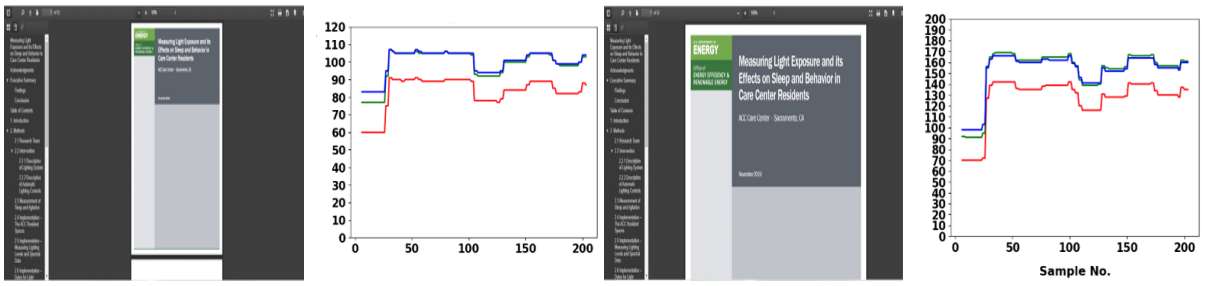
-Zooming the screen
-Action speed variation
-Action in reverse
-Random Noise

7.5.1 *Zooming the screen*

Zooming users screen may record different patterns, as they cause variation in screen contents', their shapes and sizes. However, based on our observation, varying zoom settings while reading have not generated any new pattern (Figure 7.9).

7.5.2 *Action speed variation*

Different actions can take place at various paces, based on different time and user. We vary the playback speed of actions and analyse patterns. As observed, RGB fluctuations within a specific timeframe become faster/slower, which significantly differs than original speed version.



(a) (b) (c) (d)

Figure 7.9: Analysing effect of zooming with various page setup (a) 50% zoom, (b) RGB readings with 50% zoom, (c) 100% zoom, (d) (b) RGB readings with 100% zoom. x-axis: Sample no. y-axis: recorded values

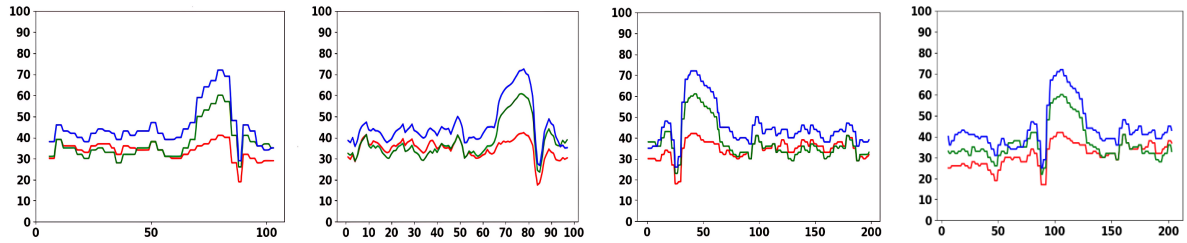
We try to re-generate alternate patterns by resampling normal speed data. Figure 7.10 shows an example, where we observe real-world faster playback containing minute variations which the *scipy* library generated version fails to exactly replicate. However, it captures the overall pattern information.

7.5.3 *Action in reverse*

Screen actions may not follow the same sequence of the data acquisition. That is why we observe what happens when the action takes place in reverse order, which is the highest level of dissimilarity. To replicate reverse action, we inverse samples using *python* library and compared it with playing the action in reverse mode.

An example has been depicted in Figure 7.10, where we played the same video in the prior example in reverse. Observation reveals library version can nearly replicate real word reverse playback. We implement and generate action speed, action in reverse, and random noise variations of our previously collected dataset and include them into the training set.

After collecting original and real world variation data set, we study several *ML, NN, and Time Series* based classifiers. For ML based classification, we select Decision Tree (DT), Random Forest (RF), K-Nearest Neighbour (KNN) and Support Vector Machines with radial (SVM-Rad) and polynomial (SVM-poly) kernels. However, with dataset having small set of examples with high variation, classifiers can behave as weak learners and may tend to overfit. To improve prediction in general, we introduce ensemble based boosting algorithms and select Adaboost (Ada) and Extreme Gradient Boosting (Xboost) for classification. To extract diverse features from our time-series observations,



(a) (b) (c) (d)

Figure 7.10: Comparing real examples with library generated examples. (a) Playing a video at a faster speed, (b) Resampling with *scipy*, (c) Playing the same video in reverse, (d) Rearranging samples with *Numpy*. Although library generated examples miss minute details, overall pattern remains almost the same. x-axis: sample no. y-axis: recorded values

we use Random Convolutional Kernel Transform (Rocket) and its faster variant Minimally Random Convolutional Kernel Transform (MiniRocket). As signals from BLE device is basically 3-channel time series data, we have included both Neural Networks (Feedforward Multilayer Perceptron Models (MLP), Convolutional Neural Networks for classification. In activity data, we expect some underlying relationship among samples in a particular activity. For that, we have also included Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) for classification.

We scale, normalize and divide RGB dataset into training, test, and validation sets (80%, 10% and 10% respectively). For better evaluation of the built model and to ensure representation from each group, we have implied *stratified 10-fold cross validation* by tuning to their best hyper parameters using *Gridsearch*. As observed (Figure 7.11), the overall classifying performance of ML algorithms are better than NNs in the controlled scenario. The first and second best results are achieved with *KNN* (**mean accuracy: *97.67%, F1 score= 0.89***) and *Xboost* (**mean accuracy: *97.25%, F1 score= 0.88***).

7.5.4 Analysing Mis-classifications

After slicing the observation into 25 RGB-C sample sub-sequences, we apply our best performing *KNN* classifier to identify the activity type that those sub-sequences represent. We analyze some of the misidentifying examples from random test and with the highest performing classifiers. As discovered from confusion matrices (Figure 7.12), some of the wrong predictions display certain patterns. For example, when the screen was on screensaver *Ribbon* mode, our classifier got

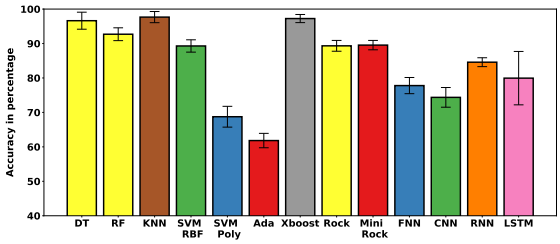


Figure 7.11: Variation of *Mean Accuracy* along with *Standard Deviations* for tuned classifiers. As seen, *KNN* and *Xboost* triumphs for generating highest accuracy with least deviation combination among all

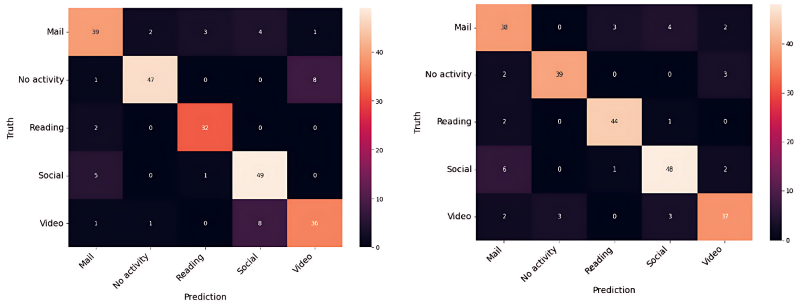


Figure 7.12: Confusion matrices for random test examples for *KNN* (left) and *Xboost* (right)

confused because of its similarity with video activity (Figure 7.13). Despite RGB pattern similarity, the brightness of the screen is different when the screen is on screensaver mode and while it is running a video. For that, we have included clear data of the observations, which is proportional to intensity information and re-train our classifier from scratch. As seen from Figure 7.19, still *KNN* ***(accuracy: 98.875%)*** and *Xboost* ***(accuracy: 98.91%)*** were the best performers with elevated accuracy. We proceed with our analysis with RGB and Clear component and with *KNN*, considering ease of real world implementation on edge devices than *Xboost*.

7.6 Realistic Dataset Augmentation

After training our classifier with a diversified database under controlled scenarios, we shift our best-performing classifier in realistic scenarios. Based on our inspection of several testbeds, we monitor that even after training the best model with fine-tuning and wide-ranging examples, performance has deteriorated substantially. Later, we address three major mis-classification incidents. The incidents, along with the approaches we take to encounter them, have been discussed below:

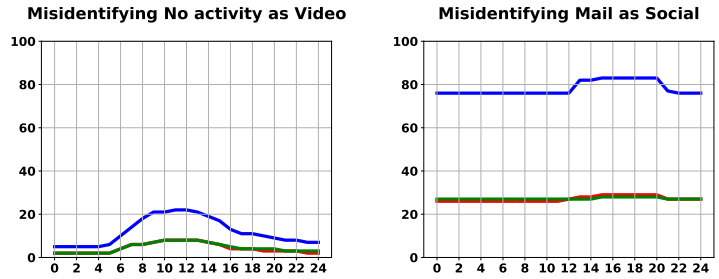


Figure 7.13: Mis-classifying examples with KNN classifier, x-axis: sample no. y-axis: recorded values

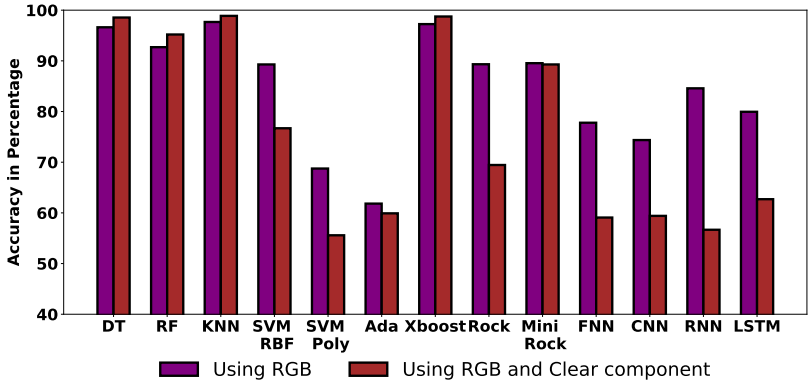
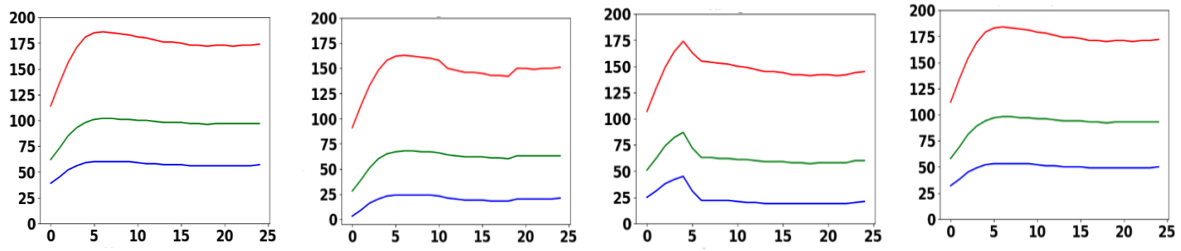


Figure 7.14: Recalculated Accuracy after adding Clear Component. As observed, accuracy has increased for ML based classifiers, especially for both *KNN* and *Xboost*, while it has decreased for the majority of the NN based classifiers

7.6.1 Influence from Indoor Light Sources

Common indoor sensors capture both RGB and the Clear component, which is the unfiltered version of light and represents the brightness information of the source(s) in the surroundings. In the presence of ambient light, there is an interaction in between indoor light and the light coming from the screen. The resultant can differ significantly from the version which is originating solely from the screen, leading to incorrect predictions (Figure 7.15). Schwittmann et.all. [77] did similar work with background lighting, but only with the single resultant at the sensor point with a single type of light. However, in the real world, sources can be of multiple types and the resultant at the sensor, however, depends on multiple measurements: (a) the relative distance between two sources, (b) the positioning of the sensor. To enable activity classification under light, we collect samples from 5 different types of major indoor artificial lights from 3 different categories: *LED, Inc and CFL* and natural light at different conditions *(Morning, Noon, Evening, Rainy Day, and Overcast). In all scenarios, color parameters recorded only with light sources were significantly higher in magnitudes than the activity values, to simulate the real-world scenarios*. We mix data from light



(a) **(b)** **(c)** **(d)**

Figure 7.15: Analysing activity under light (a) *CFL ("natural daylight", 13 W)*, located 150 cm screen, (b) Mailing (c)Reading, (d) Watching Video under light. As seen, finally recorded values were dominated by the light source, where RGB influence from video activity was the least.

and screen to mimic scenarios under the light. The highest possible deviations can be for constructive and destructive interference, where the resultant patterns were significantly dissimilar from the darkroom setup. For that, *we have considered both extremes by adding and subtracting signals from Light Source and Screen, which represent the highest possibilities for misclassification.*

7.6.2 Switching between Activities

Within the sampling window, switching from one platform to another or sometimes frequent movement within the same activity generate transitional patterns that are unalike signals captured without switching. These patterns can occur randomly and for an unknown duration and classifiers tend to misidentify when asked for identification. While moving back and forth within a single activity (like *closing an email and opening another*), we want our classifier not to get perplexed. For windows containing multi-activity observations (like *movement from Mail to Video and coming back to Mail*), we want to tag such time frames with the activity that contains the highest number of points within those time frames. For that, we have designed and developed filters for specific variable size window. When applied to regular patterns, these filters are capable of mimicking real-world transitional scenarios. For multi-actions scenarios, we implement these windows in a primary action and replace those points with a secondary action. We keep in mind that the secondary action duration never surpasses the primary one within that particular timeframe and label the frame as the primary category *(like 15 samples of yahoo mail and 10 samples of a video*

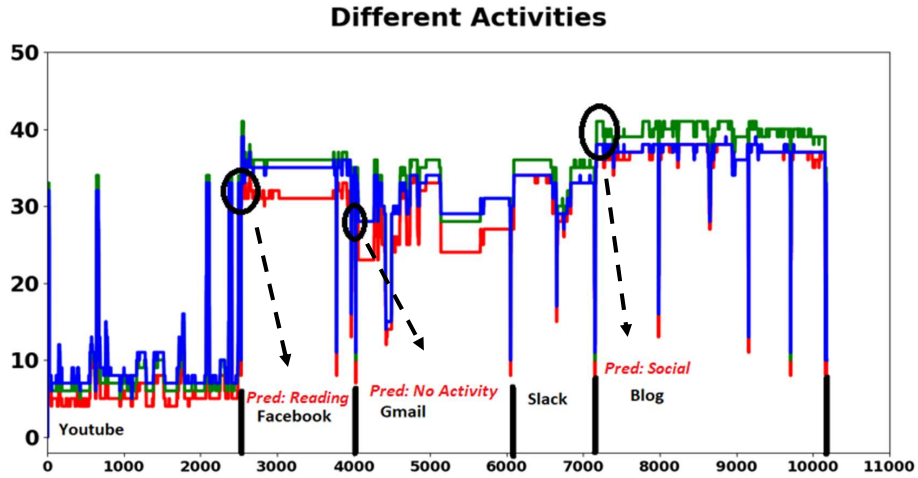


Figure 7.16: Wrong predictions events for during roaming around multiple platforms.

song, will be labeled as "Mail") (Figure 7.17).

7.6.3 Unfamiliar Activity Pattern

With a limited dataset, machine learning models are prone to overfitting, making universal classification problematic. It is also impractical to train models on all possible activity variations a user might encounter. Therefore, we include synthetic examples in our dataset, generated based on the original data distribution, to represent the absent real-world scenarios in the current training set. While there are various methods to generate synthetic data, our time-varying RGB dataset includes both static (e.g., constant values during screensaver mode) and dynamic (erratic variations) features, which need to be considered. To tackle the complexity of generating time series data with intricate temporal dependencies, we employ the TimeGAN architecture. This approach captures irregular

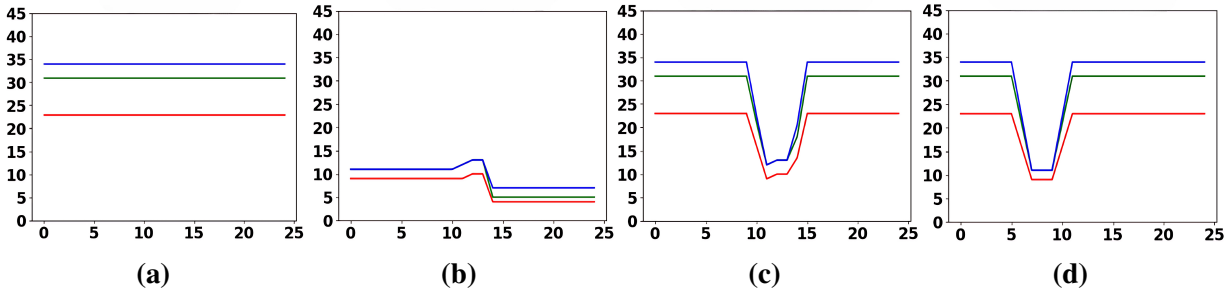


Figure 7.17: Mixing original (a)*no activity* window with interfering (b)*video data* window. (c), and (d) exhibit addition of interfering signal randomly (in a 5 sample point duration) to mimic multi-event window in real world.

and random fluctuations, as well as constant attributes, providing more challenging examples for models compared to simpler methods like autoregressive models *(details have been discussed in*

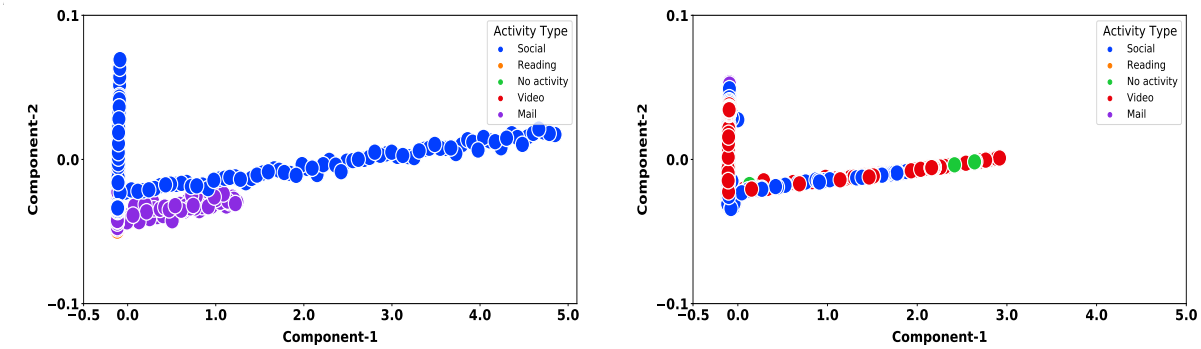


Figure 7.18: Comparing first two principal components of Captured (left) and TimeGAN generated examples (right). As observed, maximum variation occurred in *social platforms*, as it contains videos, images and texts

7.6.4 Influence from nearby screen

We place our sensor in multi-display environments, placing displays side by side, and observe RGB variation of the primary screen. In practice, the influence will depend on the positioning of the sensor relative to the secondary screen and its orientation towards that screen. As we point the sensor towards the primary screen, we observe that the primary signal amplitude varies randomly within 2%- 5% in the presence of a background second screen. To model this effect, we add a random RGB noise (5% of components amplitude) to our dataset.

7.6.5 Performances at Realistic Testbeds

We conduct three experiments at three different realistic testbeds (Figure 7.20). They include samples taken in a dark room with unseen user, not included in the training set (*Testbed-1 (top)*), under unknown room light single display with a completely unseen screen settings (*Testbed-2 (middle)*), and unknown room light multi-display scenario (*Testbed-3 (bottom)*). All testbeds contain five activity types for classification, with the first having seen the content and the others having unseen content with practical variations *(zooming, varying playback speed)*.

Following the segmentation of the observations into 25 RGB-C sample sub-sequences, we employ our top-performing *KNN* classifier, which has been trained using RGB and Clear components in

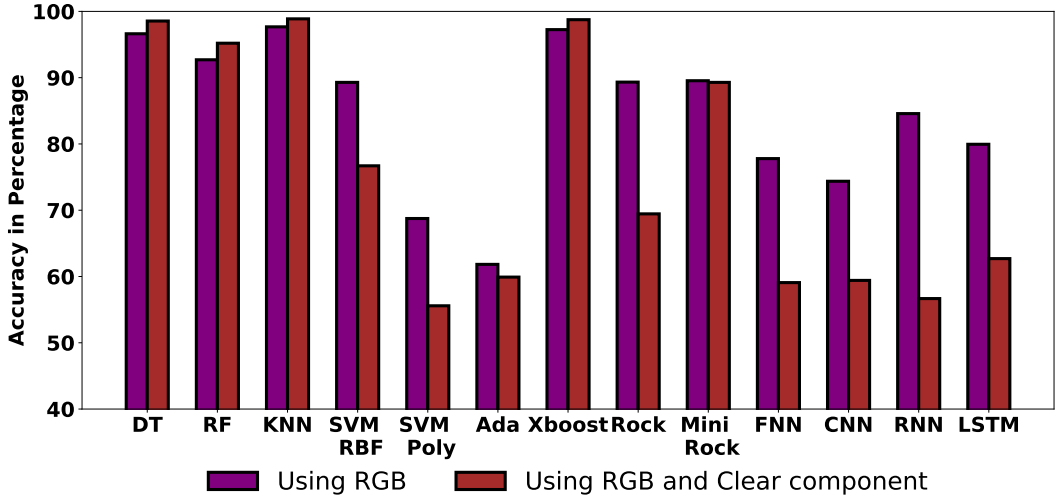


Figure 7.19: Recalculated Accuracy after addition of brightness information.

controlled scenarios, to determine the activity type represented by these sub-sequences. We observe that the classification accuracy has been degraded significantly in all the testbeds (*as low as **72.5%***). In the dark room scenario, the classifier failed to identify timeframes where the user was switching from one activity to another. At other testbeds, the classifier simply gets confused with patterns under light and practical variations of activities.

7.6.6 Performance after augmentation-retraining

We then add augmented data in the present training set and retrain our *KNN* classifier with this augmentation. Accuracy of *KNN* classifier pre and post-implementation of augmented data were recorded. As seen from Figure 7.21, in all three scenarios, performances were elevated (***upto 7.5%***) with the extraneous training set, and the with highest accuracy recorded was for *Testbed-1 (**accuracy: 91.25%, F1 score= 0.81**)*, compared to test set accuracy of ***79.3%*** with Random forest and ***70.1%*** with Naive Bias reported in [33]. Figure 7.22 (top) depicts a few examples where the introduction of extraneous training sets aid in accurate predictions of activity class, which were previously misidentified with the limited training set. However, there were a few timeframes that were still misidentified (bottom). After analysis, no common patterns of inaccurate predictions were observed.

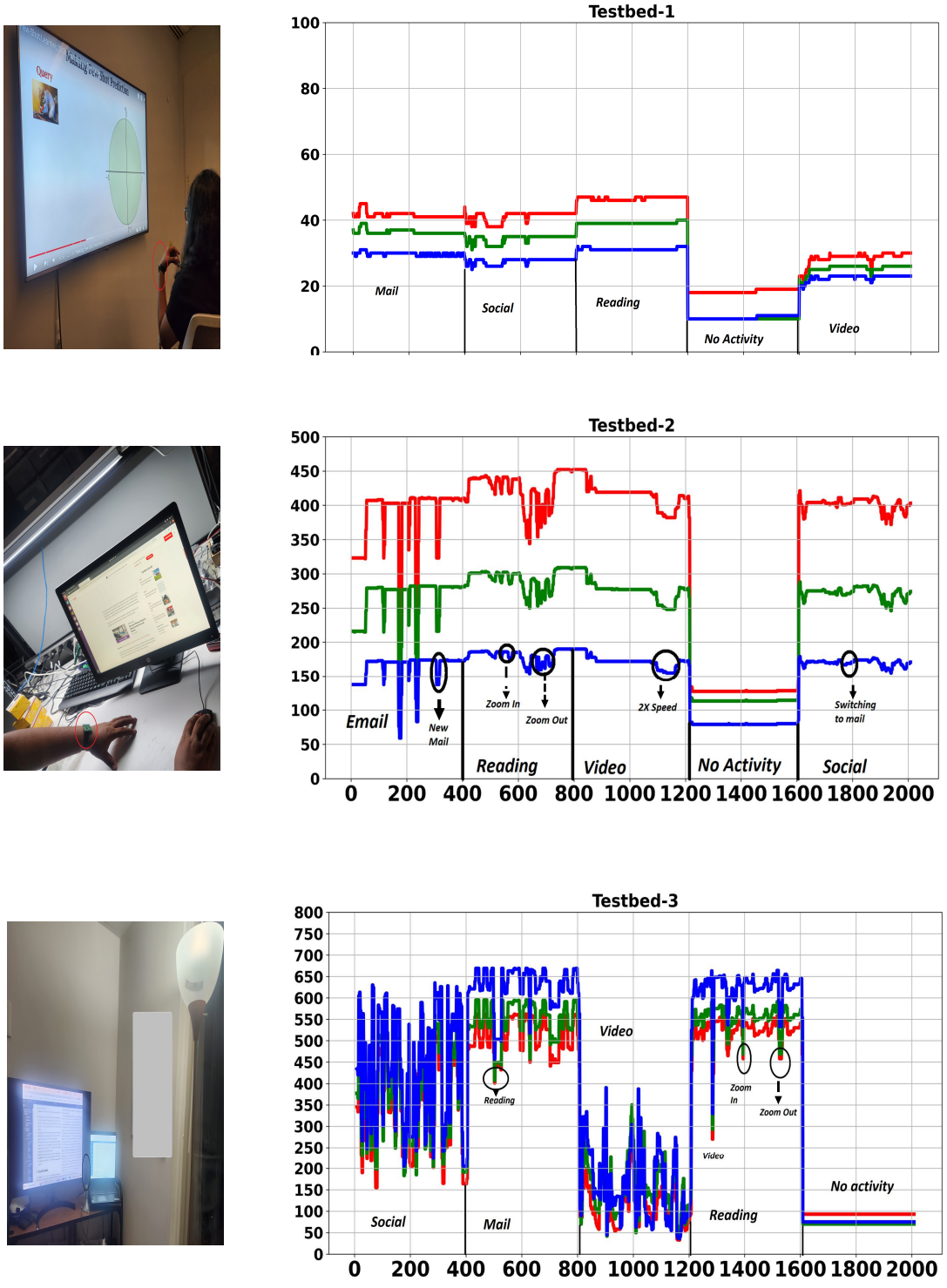


Figure 7.20: RGB signals at different test beds from screen activities. Each testbed comprises of 2000 samples including five activities with realistic variations. Collecting signal: (top) as wearable at dark conference room at *1m* distance from *LG 55" screen with custom setting*, (middle) as wearable in a lab with *HP 32" monitor* under *LED: 21W, 3000K* (middle), as fixed point device with *LG 34" & Ideapad S145 monitors* under *LED: 65W, 5000K* lamp (bottom)

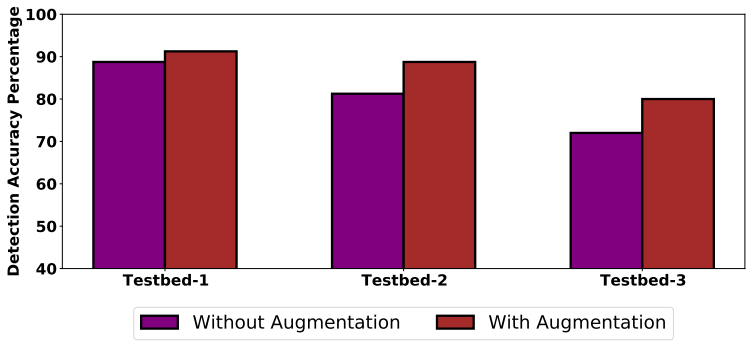


Figure 7.21: Comparing Testbed accuracy with training with and without augmented dataset

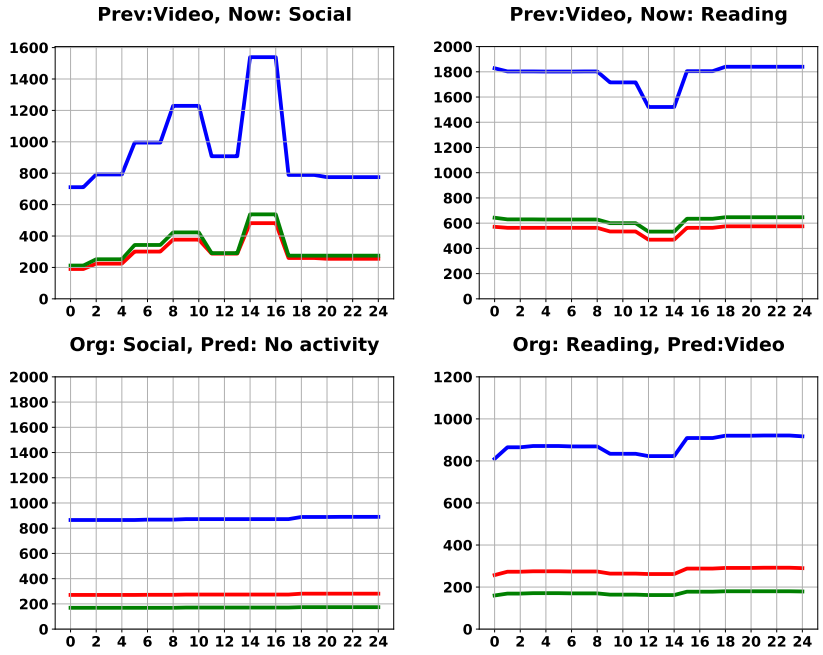


Figure 7.22: Correct Predictions with extraneous training set (top), Miscategorized examples recorded after retraining with extraneous set (bottom) (x-axis: sample no., y-axis:recorded values)

7.7 Discussion

Data acquisition setup: For Apollo, we set *TCS3475* sensor with 1X ADC gain with an integration time of 700 ms, which allowed us to read color values up to value 65535. For low light conditions or activity detection beyond 1 m, tuning ADC gain settings achieve this, though with the trade-off of higher energy consumption. The average current consumption can be reduced by reducing the sampling and transmission frequency. However, the frequency should be high enough to capture enough variations and enable real-time data acquisition.

Performance limiting factors: With Apollo, we have not considered what happens when the

sensor is positioned at mid-point and equal distance from two displays. We assume that while carrying as a smart device, user hand was not in motion, which can impact the final recordings. With low RGB amplitudes from screen, too bright room light in many scenarios may entirely overshadow on-screen information. Apollo can be arranged to be auto-synchronous with the active screen. However, additional arrangements may be called for to ensure users' actual involvement on a particular screen, like gaze control or face detection arrangements [78], [79], as the user may easily switch in between monitors or looking places other than monitor in realistic scenarios. Our training set includes a limited type of screen setting variations. For customized/specific settings, the performance of the classifier can be elevated by introducing few examples in the training set with that setting. With limited examples and features, ML algorithms have outperformed neural networks, but as the dataset and features expand, neural networks may surpass traditional ML algorithms. Finally, in real world, screen usage is highly dynamic, which demands augmentation/modification of the number of categories.

Chapter 8

SENTREC: ROBUST AND DATA-EFFICIENT INDOOR EVENT IDENTIFICATION WITH PHOTOVOLTAIC CELLS

On one side, the capacity of the world's photovoltaic (PV) systems is experiencing unprecedented growth; on the other side, the number of connected devices is rapidly increasing due to the development of advanced communication technologies. These fields converge, with recent studies identifying photovoltaic energy harvesting as a promising green solution to the energy challenges of next-generation IoT-based applications in smart homes, cities, and factories [80]. The size of the indoor solar cell market was valued at USD 81.1 Million in 2023 and is projected to reach USD 154.7 Million by 2031, growing at a CAGR of 9.6% during the forecast period 2024-2031 [81].

In indoor environments, PV systems provide several advantages over conventional power sources as energy harvesters. These sensors are flexible to be deployed in remote and hard-to-reach spaces, offering high power conversion efficiency, flexibility, and exceptional specific power performance [82]. Fascinatingly, the harvester's voltage output fluctuates as individuals move, causing variations in the light reaching the cell. This led to a key insight: changes in illuminance levels caused by specific external activities leave distinct imprints on the voltage patterns of the photovoltaic (PV) cell, effectively encoding information. For example, when placed near an entrance, the voltage variations caused by a person walking past a PV-powered sensor are uniquely identifiable, reflecting individual differences in height, body shape, and gait. Intriguingly, these patterns also exhibit a directional characteristic. This explores the multimodality of these specific types of indoor light sensors, both converting light into power for indoor devices and serving as indicators of specific events, such as exit / entry or identifying individuals involved in that particular event [83].

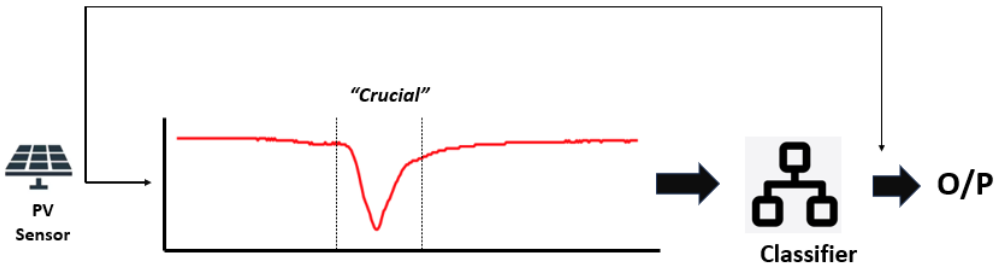


Figure 8.1: Offloading computationally intensive exit/entry event classification task to a remote device

Let us consider a scenario where a photovoltaic sensor can indicate an exit/entry event through variations of recorded voltage values. A classifier is trained on a distant platform to identify the event based on multiple similar observations (Figure 8.1). However, as observed, the perturbation during the event is only briefly observed. An energy-efficient way of operation includes the sensor remains active only during that perturbation and remains silent/sleeping mode during the steady-state period (Figure 8.2). However, achieving this requires re-designing the operability of the sensing system in a smart manner. This includes identifying and filtering out the irrelevant information, and transmitting data related to specific event perturbation. Additionally, the remote classifier should be pre-trained to accurately detect short, distinctive patterns corresponding to that particular event.

8.1 Limitation of Data Efficient PV based Event Sensing

While existing research has explored techniques for efficient event classification in long recordings, such as compact sensing, data encryption, on-device prediction, and intelligent event detection, a critical gap remains. Existing approaches often neglect crucial practical considerations. Specifically, *how can we effectively classify specific events while simultaneously disregarding other events that*

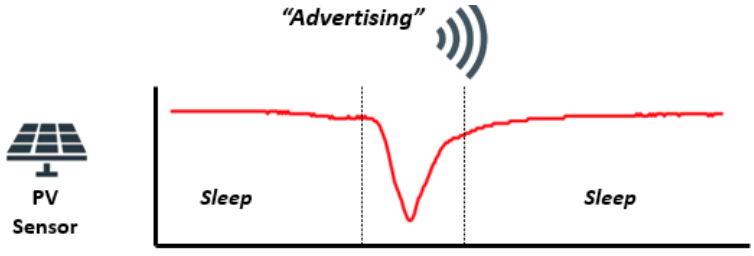


Figure 8.2: Energy efficient operation of PV sensors

the sensors may also detect? Classifying non-target events consumes unnecessary resources and is undesirable. By selectively transmitting only targeted events, particularly in scenarios where non-target events occur more frequently, we can significantly reduce energy consumption at the sensor node. Furthermore, *how can we ensure the robustness of the transmission process in environments where the sensor is sensitive to nearby interference or the characteristics of the received signals may change in a dense sensor deployment scenarios?"*

8.2 Challenges

Developing robust and efficient platforms for classifying specific indoor events presents multiple challenges, including sensor variability, environmental noise, and the complexity of identifying relevant event-related segments:

- Effective classification requires distinguishing between variations caused by targeted events and non-target events/factors while optimizing segment selection for accuracy and energy efficiency. For that, we need to innovate a strategy that can differentiate between them and easily deployable in low power systems.
- The length of perturbation can vary from one event to another. Therefore, choice of segments can be tricky. Longer segments can waste memory, while shorter ones risk reduced accuracy.
- In real world, we would like to select segments that are the most confident and useful for classification. However, the segment with the highest confidence not may not tbe the most robust. Therefor, the selection should demonstrate both confidence and exhibit robustness to real-world interference is critical.
- However, identifying robust segments is challenging, especially for non-gradient-based classifiers, as no universal methods currently exist.
- Additionally, IoT-based sensors face constraints in energy consumption, requiring efficient data compression and minimal transmissions for classification. Real-time segment selection further complicates implementation on low-power embedded sensors, demanding innovative, resource-efficient algorithms that outperform traditional statistical methods.

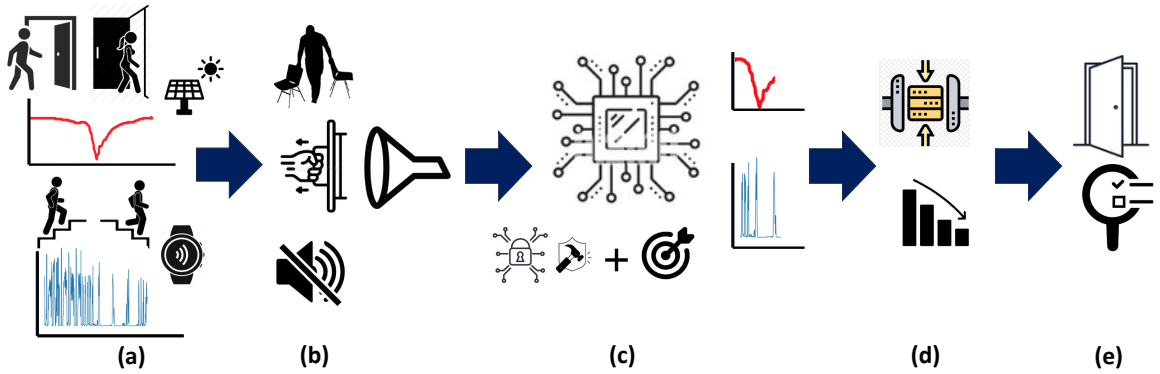


Figure 8.3: Robust and Data Efficient Roadmap: (a) Selecting indoor events *(like, occupant detection/ walking upstairs-downstairs)* from Solarwalk and UCI-HAR dataset for analysis, (b) Filtering unwanted chair/door dragging events or environmental noise for classification that create similar perturbations at the sensor nodes, (c) Picking up specific segments based on robustness and accuracy, (d) Identifying appropriate and edge device friendly lossless compression techniques to lower the number of advertisement packets, (e) Implementing proposed approach under realistic testbed for evaluation

8.3 SENTREC: our proposed platform

To address these limitations, we introduce SENTREC, an innovative platform designed to detect targeted indoor events with enhanced data efficiency, as well as improved robustness. SENTREC filters out non-target events from classification, leading to energy savings in both data transmission and processing. Furthermore, SENTREC utilizes a smart algorithm to detect and transmit only the essential segment needed for classification, avoiding the transmission of extended sequences. This smart algorithm also enables SENTREC to select segments based on the combination of confidence and robustness. This ensures the classification process remains resistant to surrounding noise, which could otherwise significantly degrade performance. Furthermore, SENTREC utilizes a smart algorithm to detect and transmit only the essential segment needed for classification, avoiding the transmission of extended sequences. This smart algorithm also enables SENTREC to select segments based on the combination of confidence and robustness. This ensures the classification process remains resistant to surrounding noise, which could otherwise significantly degrade performance.

Depending on the type of data, SENTREC can effectively compress the information required for classification, enabling offloading with significantly lower number of advertisement packets. This

allows the sensor to operate for extended periods without requiring a battery replacement or under limited power supply conditions. Finally, when deployed in a real-world environment, SENTREC outperforms traditional statistical based methods in processing streaming sensor data. Figure 8.3 describes the overview of our proposed architecture.

8.4 Related Work and Limitations

In this section, we discuss related approaches and their limitations in identifying indoor events.

8.4.1 Identifying Event Segment

Hanlon et. al. proposed a real-time gait event classification through voltage threshold setup at multiple levels [84]. However, the threshold-based approach may prove inefficient, as non-targeted events are observed to create similar variation. IRESE, a real-time rare-event detection system, uses unsupervised machine learning to analyze incoming data [85]. However, its reliance on complex feature-based micro-clustering makes it unsuitable for energy-constrained sensors. A novel wireless indoor event detection system TRIEDS was proposed by Xu et.al. which uses time-reversal techniques to capture channel state information (CSI) changes to detect multiple events [86]. The platform is vulnerable to environmental noise and multipath interference, which can impact the accuracy and reliability of event classification. Luo et al. proposed an event detection system powered by harvested energy that adjusts its sleep cycle based on event arrival patterns [87]. In real-world settings, indoor events, such as occupant movement, often occur in a highly irregular manner. So, synchronizing device on-off based on prior incidents can be problematic and inaccurate.

8.4.2 Energy Efficient System Design

To establish systems balancing energy optimization and user privacy, Errapotu et.al. proposed scheduled and encryption based approach [88]. Encryption can enhance smart home security but introduces potential performance and implementation challenges, especially in complex setups. Karjou et al. proposed a low-cost, privacy-friendly way to measure occupancy using low-power wireless networks and cloud processing [89]. This scalable approach, while ensuring consistent and cost-effective performance through data integration, may be impractical in single sensor scenarios.

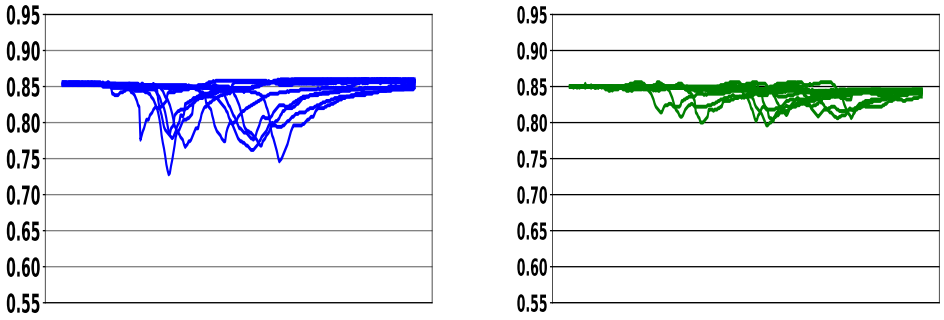


Figure 8.4: Variations in sensed values recorded during 10 different instances over time: PV variations in voltage measurements for movement through the doors for occupant-1 (top-left) and occupant-2 (top-right).

8.4.3 Robust Classification

On device classification of events have been proposed for Memory-Constrained Intelligent by Kim et.al [90]. Even though architecture can identify out-of-distribution examples, on-device classification requires extensive memory and processing power for universal use. Abououf et.al. proposed (LSTM) autoencoder for real-time anomaly and event detection, integrating smart inference to optimize power consumption and extend device lifetime. Due to dependence on cloud-based training, the approach leads to latency issues and reduce real-time adaptability. To ensure robustness, SHA3-256 hash function was used by Jan et.al. [91]. The protocol uses too many extra bits, leading to communication overhead.

8.4.4 Efficient Compression

A combination of Autoregressive Prediction and Huffman encoding has been proposed by Ahmed et al. for data suppression [92]. These approaches are not be suitable for activities that possess both static and dynamic features. Lossy compression techniques, like K-run-length encoding (KRLE), lightweight temporal compression (LTC) etc. can indeed achieve a higher compression ratio [93]. This approach, however, incurs an accuracy loss, making it unsuitable for systems with performance thresholds. Halah et al. demonstrated that advanced lossless compression schemes such as S-LZW and S-LEC achieve higher compression ratios [94]. However, sensors with limited power or memory require simpler compression algorithms with smaller code sizes.

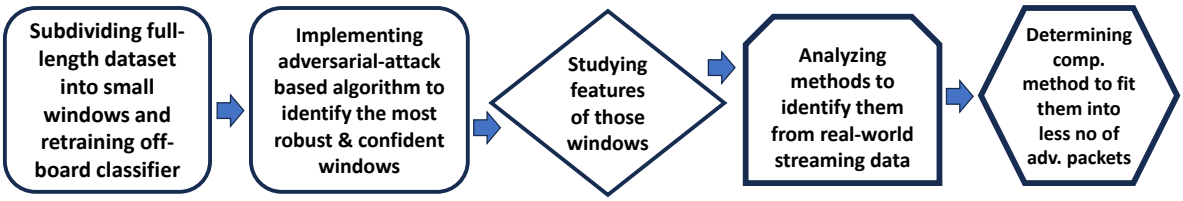


Figure 8.5: Workflow for SENTREC architecture

8.5 Selected Dataset for Experimentation

The dataset includes variation in photovoltaic traces affected during human movements. It includes observations collected from five individuals with varying BMI in two separate rooms within a building. Saoda et.al. recorded 150 minutes of events corresponding to a total of 900 door entry and exit walk events. PV cells were placed at an optimal point and under LED lights. The platform captures the voltage variations due to movements and streams data using the MQTT protocol to a cloud-hosted database. Each example consists of 300 voltage values, collected at 50Hz sampling rate [95].

8.6 Methodology for implementation of SENTREC System Architecture

As depicted in Figure 8.5, our proposed architecture for SENTREC consists of six key stages, ranging from segment re-design to final compression for classification. Figure 8.4 presents several examples of variations across different classes in the aforementioned datasets. However, the perturbations caused by an occupant's movement were not time-synchronized.

8.6.1 Optimal window selection

Our goal was to determine: *(i) how the overall accuracy varies across different window segments during the entire event, and (ii) the accuracy attainable with a specific window size.* For that, we **subdivide the original time series data into multiple equal-length sub-sequences using overlapping sliding windows of varying lengths**. After dividing the windows into random training and test sets for windows in every position, we then retrain the reported best performing KNN classifier with the training set and record the performances with the test set for variable-length

windows. We analyze the accuracy variations across different positional windows and also record the peak accuracy for a particular window size. Finally, we identify the optimal window size for each dataset based on the mean test set accuracy, which we later use for advertisement and re-training the best reported classifier off-board.

8.6.2 Analyzing confident windows

To pick up the window with the highest confidence, we observe *how the confidence in the correct class fluctuates across these windows for a particular observation* and *whether the choice of the most confident window is unique or not.*

8.6.3 Adversarial Attack for Robustness Analysis

Adversarial noise in IoT sensor networks can generate disturbances similar to environmental interference or sensor malfunctions. It can introduce random variations in time series data, such as voltage or motion readings, which resemble natural sensor noise or interference from nearby devices in crowded sensor environments [96]. For that, to identify the most robust sub-window within an observation, we utilize the **Iterative Fast Gradient Sign Method (I-FGSM) attack**. This attack generates adversarial examples by iteratively applying small perturbations in the direction of the gradient of the loss with respect to the input, gradually increasing the input's deviation until it misleads the model (details in [97]). I-FGSM is superior for real-world implementation due to its simplicity, speed, and efficiency. It utilizes a single gradient-based step compared to the more complex, iterative, and computationally expensive Projected Gradient Attack (PGD) [98] or Carlini & Wagner (C&W) attacks [99], making it ideal for rapid analysis of time series based observations.

8.6.4 Universal Robustness for Gradient & Non-Gradient Classifiers

One of the major limitations of implementing I-FGSM universally is that it relies on the gradient of the loss function with respect to the input to generate adversarial perturbations. When the best-performing classifier is non-gradient-based *(e.g., K-Nearest Neighbor, which has shown superior performance in identifying occupant based on PV variations but does not compute gradients),* applying I-FGSM directly becomes challenging. To address this, we generate a **surrogate model** that achieves similar accuracy to KNN. While various methods like Zeroth-Order

Optimization (ZOO) and Simultaneous Perturbation Stochastic Approximation (SPSA) exists for generating gradient-based attacks on non-gradient-based classifiers, we prefer surrogate models as they effectively approximate the target classifier's decision boundaries [100].

As models with closely aligned decision boundaries are likely to exhibit comparable classification behaviors on similar input data, we test several neural network models to match KNN's performance [101]. However, due to the small size of the occupancy detection dataset (only 900 examples), gradient-based neural network models find it challenging to outperform KNN. As occupancy detection is a time series task, we find that **an LSTM (Long Short-Term Memory) network** outperforms other neural network models in this case and achieves accuracy similar to KNN. We take the *loss gradients from the surrogate LSTM model* to generate adversarial attack on the windows of occupancy detection datasets. We then feed these adversarial examples into the KNN classifier.

8.6.5 Analyzing Identified robust examples

Our next step is to make the following different queries:

- *How confident the detected windows are regarding correct class prediction?*
- *Is the maximum robustness unique for an observation?*
- Due to the overlapping samples in neighboring windows, they may exhibit similar features. This can lead to the selection of adjacent/offset windows as the desired ones for streaming data in real-time scenarios. In that case, *how confident the offset windows are?*
- *Are these windows really capable of overcoming real-world interference or not?*

To answer the abovementioned questions we do the following experiments with the test set:

- **Confidence:** *To query that, we calculate the confidence for the correct class of the detected window across different examples.*
- **Uniqueness:** *We observe the number of the steps necessary to make a successful attack for an observation.*
- **Offset:** *We assess the confidence levels of the correct class for these neighboring windows and compare them against the detected ones.*

- **Robustness:** *We generate random noise and apply it across all the windows for an event where many windows indicate the correct class with high confidence. Then we evaluate how this impacts confidence in the correct class prediction for the detected window versus the other windows.*

8.6.6 Comparison among approaches for optimal window selection

During an event, sensed values deviate from steady-state patterns, displaying distinct statistical features. These features can serve as indicators of the occurrence and type of specific events. Selecting windows based on indicative statistical features is straightforward to implement on-board and can effectively facilitate targeted window selection. To achieve this, windows are aggregated based on specific statistical feature across all observations, and their overall accuracy is assessed in comparison to the most robust windows identified through our proposed method.

8.6.7 Window selection for streaming data

We aim to implement real-time methods for identifying desired windows from streaming sensor data, while efficiently filtering irrelevant events. Given the limited scope of this paper, **we have chosen to focus specifically on occupant detection,** concentrating our forthcoming approach on this aspect. However, this approach can be generalized to other datasets as well.

We explore three methods to identify the most robust windows in streaming data. For primary evaluation, we consider 300-sample voltage traces from Solarwalk test dataset as real-world streaming examples before moving to realistic testbeds. We then select the windows that meet specific robustness and confidence criteria for each method and save them for each observation. Our next step was to analyze their features and separate them from other windows within an event. For that, we consider three different approaches as discussed below. Finally, we deploy the optimal approach on-device in testbed, chosen based on their performance and resource-friendliness for on-device implementation.

8.6.8 Pattern based approach

A robust and confident reference window from the Solarwalk dataset, excluded from both training and testing, is selected for pattern matching. The idea is that robust windows from various observations might exhibit similar patterns. We then **calculate the Dynamic Time Warping**

(DTW) distances of each window in each example for similarity measurements, with lower distances indicating higher similarity. The rest of the process works as follows:

- For each example, we record the window with the minimum distance (or highest similarity) and plot the distribution. Based on the distribution, we set a threshold.
- In real-time, **the first window that generates a DTW distance below this threshold** will be allowed to be advertised.
- After advertising a single window, it will anticipate the next window that meets the criterion.
- If none of the windows satisfies the criterion, it will send **the last window of the observation**.

8.6.9 Threshold based approach

We compute **the statistical features of the identified robust windows and select the top feature based on test set accuracy.** Next, we continuously adjust that feature's threshold and record the criterion that satisfies **the maximum number of most robust windows.** To mimic the real-time streaming scenario, we consider **the first window** that meets the criteria, disregarding the others. If no window satisfies the condition, we select the last window as the most robust by default. Finally, we record the classification accuracy based on the chosen windows.

8.6.10 Machine Learning Classifier based approach

The technique works as follows:

- We identify the most robust window and assign it as Class 0, avoiding any overlapping windows.
- The remaining windows are categorized as Class 1. We then select an equal number of windows from there to balance the number of examples with Class 0.
- The total dataset is then randomly divided into a training set (80%) and a testing set (20%).
- Finally, we train Machine Learning models and evaluate their performance on the test examples.

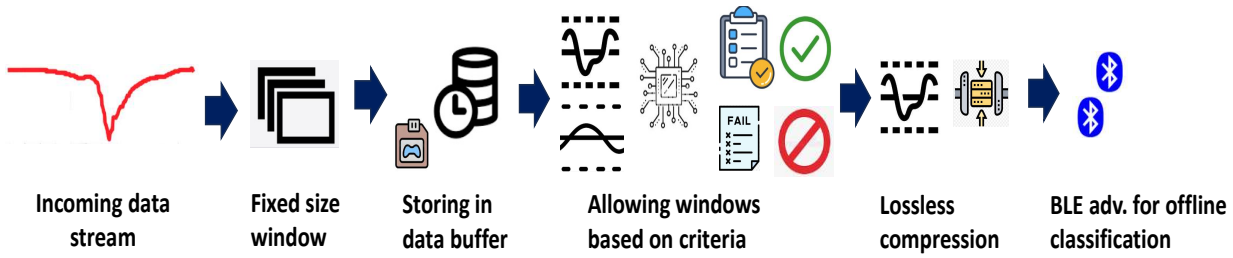


Figure 8.6: Proposed tasks to be performed on device

8.6.11 Lossless Data Compression

After selection of the optimum window, our goal is to further compress the window information, and fit it into fewer packets to minimize the total number of advertisements. To achieve this, we explore lossless data compression methods for information minimization. Considering the ease of implementation on edge platforms and the nature of our dataset, we evaluate ***Huffman, Delta-Deflate, and Run-Length Encoding*** algorithms. For Delta-Deflate compression, we first implement Delta compression on raw values and then compress the delta-transformed data using the deflate algorithm.

The proposed on-device architecture is shown in Figure 8.6. After storing the streaming data in predetermined buffer window, our developed filtering technique only allows certain window for lossless compression. The compressed data can then be transmitted through methods like BLE advertisement into fewer number of packets. Windows that do not meet the selection criteria are not considered for compression and advertisement.

8.7 Robust window searching algorithm

After re-training the best reported classifiers *(Solarwalk: KNN, UCI-HAR: LSTM)* with optimal window sizes, we implement the attack on both datasets. As previously mentioned, **adversarial perturbations generated by the surrogate LSTM model are used to attack the KNN classifier for Solarwalk, while the original LSTM was used for UCI-HAR**. To generate mis-classification through adversarial perturbation for robustness, choice of attack parameter is crucial. Too much deformation at each step may indicate multiple windows as the robust ones. On the other side, too low deformation may require a vast number of steps and may result in an unsuccessful attack with a fixed number of steps. In both cases, we applied up to **200 steps,** using a perturbation factor of

0.0008 at each step.

The key steps for selecting the robust segment are as follows:

- Under a single observation, we check whether a particular window classifies the correct class.
- Record the window as the most robust one that requires the maximum number of steps for a successful attack
- If multiple windows record the same number of steps, then select the window that has the initial highest confidence for the right class prediction. **This ensures robustness with confidence**.
- If multiple windows record the same number of steps and the same confidence for the right class, we determine **the longest running sequence of those examples and choosing the middle window as the most robust one**.
- If the window predicts the right class but the attack is not successful for any of the windows, we select **the middle window as the most robust one.**
- If none of the windows of an event predicts the right class, even then we would like to send a window. For that, we select **the middle window for offline classification.**

The details of the algorithm is described for occupant detection dataset with surrogate model (Algorithm 2).

Algorithm 2 Most Robust Window Identification

Start:

- Dividing full length observations into overlapping sub-windows of particular length
- Splitting the sub-windows into random training and test sets
- Training best reporting classifier using the training dataset
- Also, training a surrogate model and tune the hyperparameters until the model achieves accuracy similar to previously best reported classifier

I-FGSM Attack:

- $\varepsilon = 0.0008$ {Adversarial perturbation} - $\eta = 200$ {Number of steps} - δ= Number of observations - ζ= Number of windows under each observation - $count$= 0 - x is a window segment from the event - x' is x after perturbation - $\mathcal{F}(x)$ = Classifier output for the window segment - ϱ = Correct class for the window - $\nabla_x L(x, y)_{LSTM}$ is the loss gradient from the LSTM model
- y, y' are the predicted classes before and after perturbation

for $i = 0$ *to* δ **do**
  for $j = 0$ *to* ζ **do**
    $count$= $count$+1
  end
  if $\mathcal{F}(x)$= ϱ **then**
    for $k = 0$ *to* η **do**

$$x' = x + \varepsilon \frac{\nabla_x L(x, y)_{LSTM}}{|\nabla_x L(x, y)_{LSTM}|}$$

$$y' = KNN(x')$$

      if $y = y'$ **then**
        x= x'
      end
      else if $y \neq y'$ **then**
        - Record η {Number of steps}
        - Record k {Window number}
      end
      if *Multiple windows have the same step sizes* **then**
        Consider $Max(|KNN(P_{right\_class|window\_j})$ as the most robust
        if *No Max* **then**
          Select the longest consecutive series of windows with the highest correct class confidence and choose the middle window of that series
        end
      end
    end
    else if $y = y'$ *for* $k = \eta$ **then**
      Choose the middle window
    end
  end
  else
    Choose the middle window
  end
end

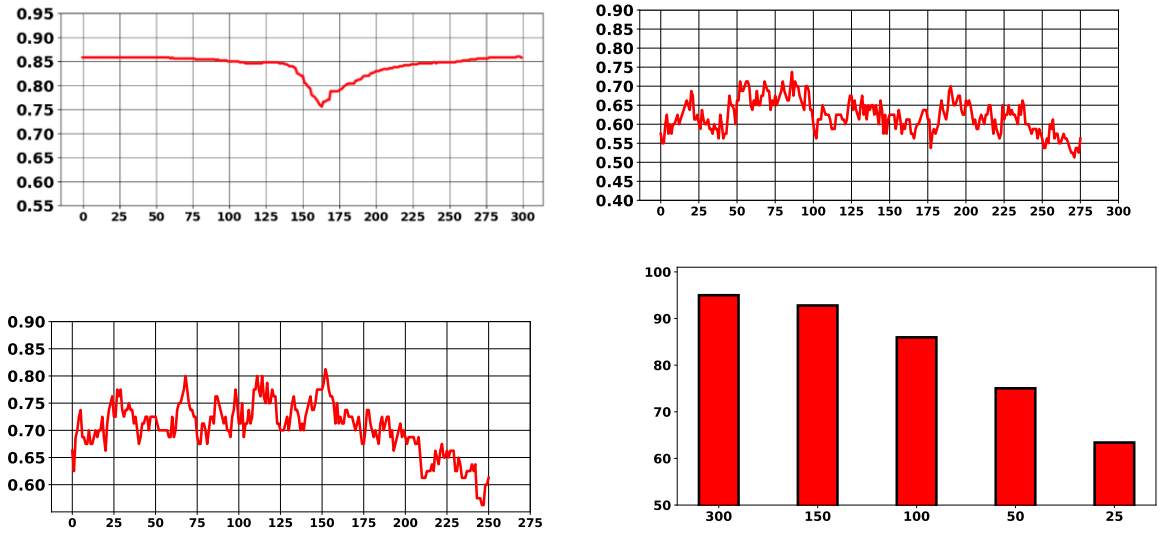


Figure 8.7: An entry event by an occupant spanning 300 samples (top-left)[x-axis: time samples, y-axis: voltage], Fluctuations in accuracy across various overlapping sub-divided windows for the PV occupant dataset are depicted: peak accuracy was observed for overlapping windows. For 25-sample windows, the peak accuracy reached 74% (top-right), while for 50-sample windows, it increased to 82% (middle-left). Mean accuracy relative to window size is plotted for Solarwalk (bottom-left) and UCI-HAR (bottom-right) [x-axis: window size, y-axis: accuracy in percentage]

8.8 Evaluation

In this section, we present the results of the experiments discussed in the design section.

8.8.1 Optimal Window Selection

While documenting variation of accuracy over different positional and size-based windows, we observe that, the accuracy keeps fluctuating at different positions for a fixed window number, especially when the even takes place in a random manner (Figure 8.7). As also observed, **the peak accuracy for a particular window has come down with decreasing window size**. To determine the optimal window size, we divide the *Solarwalk* 300 sample observations into four different window sizes: *25, 50, 100, 150*.

To trade off accuracy and data-efficiency, we select a window size of *50 samples* for occupant detection where the mean accuracy threshold of 75%. After dividing the windows of the above-mentioned sizes into random train-test sets for both datasets, **we retrain our distant**

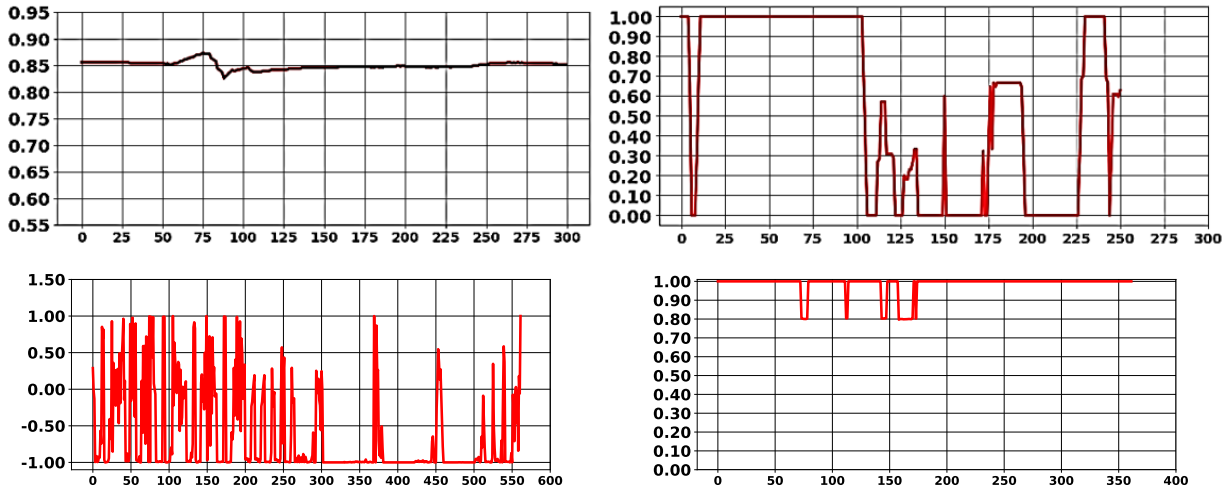


Figure 8.8: An entry event by Occupant 2 (top-left, time vs voltage) and prediction probability of different windows of class 2 (top-right, window number vs confidence). Similarly a walking downstairs event (bottom-left, time vs voltage) followed with its class probabilities (bottom-right, window number vs confidence). In both cases, multiple windows record maximum confidence.

classifier with the training set. From there, our goal is to first detect the robust and accurate window of from the test sets with redefined window size and only advertise that window for distant classification.

8.8.2 Studying Confidence

For studying the variability of confidence over different windows, we measure **the probability of the correct class** for each of the window of an observation. As seen in Figure 8.8, **multiple windows within an event** exhibit similar confidence of the correct class. In addition, within an observation, the window with the highest confidence **may not be unique**.

8.8.3 Adversarial attack for robustness

We apply a surrogate-based adversarial attack on the PV testset using the following LSTM architecture, which achieves a mean accuracy of **79.78%**, closest to that of the KNN model for 50-sample windows (maximum window accuracy of **82.0%**).

After generating attacks on the test sets fo the dataset, we analyze: the distribution of steps throughout the test set, the positioning of robust windows, and finally their level of confidence. As

Table 8.1: Description of surrogate LSTM Architecture

Layer	Type	Description
1	Input	Input shape $(50, 1)$
2	LSTM	128 units, returns sequences
3	Dropout	0.2 dropout rate
4	LSTM	64 units, returns sequences
5	Dropout	0.2 dropout rate
6	Dense	32 units, ReLU activation
7	Output	Dense layer with no of classes unit, softmax activation

shown in Figure 8.9, the level of deformation required to cause misclassification **varies across different observations, along with the indices**. On certain occasions, we observe successful attack phenomenon for a previously full-length misclassified example. This suggests that while the full-length observation may result in misclassification *(as the test set accuracy is below 100%)*, **certain windows within those examples can still correctly indicate the class for prediction.**

As observed in Figure 8.10, on many occasions, multiple windows have recorded the same number of steps for making successful attacks within an event. **As a result, only the criterion for recording most number of steps as the most robust window may not be sufficient.** The window with the highest step count may lead to misclassification when processed by the classifier. That is why in our developed algorithm, **we focus on the particular windows that yield correct class predictions, along with the number of steps for a successful attack.** In the following observations, *since the UCI-HAR dataset yields similar results, we have only presented the results for the PV dataset to avoid redundancy.* We evaluate the confidence levels for the correct class predictions across all event windows, with a particular focus on the confidence levels of the windows identified by our algorithm (based on maximum robustness and correct class prediction).

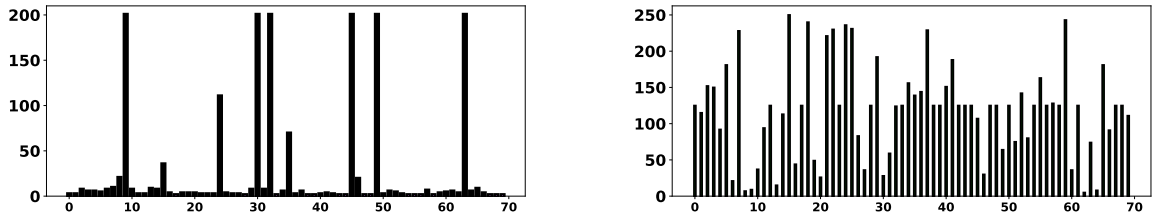


Figure 8.9: Analyzing robust examples of both datasets after adversarial attack: no. of steps required for attack all the test PV examples (left). Associated indices of the robust windows are also recorded(right)

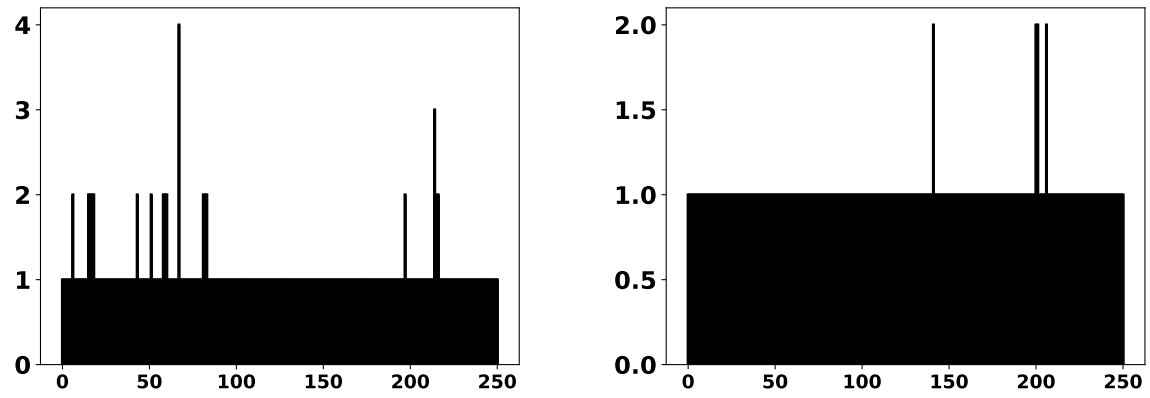
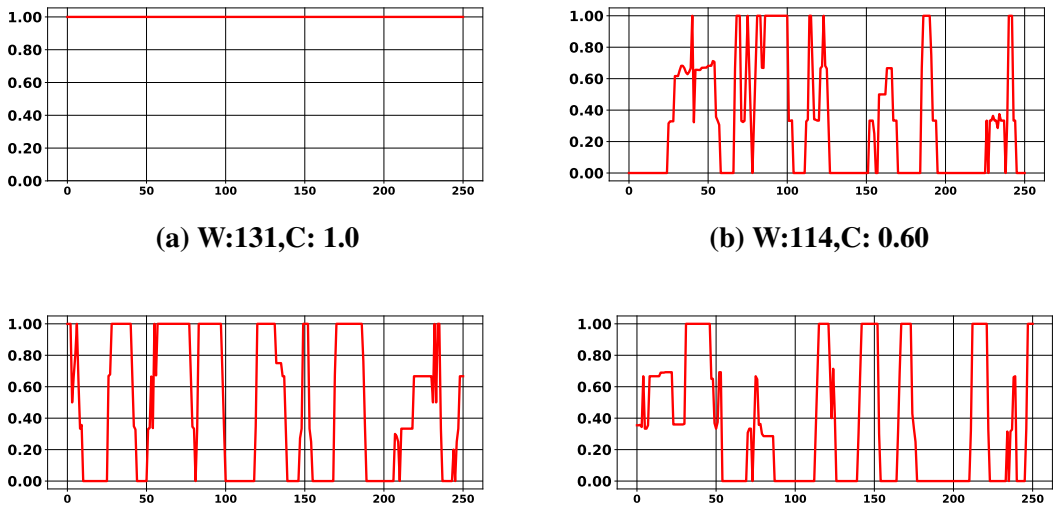


Figure 8.10: Number of steps for successful attacks for different examples on the PV test set (window number vs steps): the left example shows a unique window requiring 4 steps, while the right example has multiple windows with 2-step attacks.

As described with a few examples in Figure 8.11, **the window with the highest confidence is not always the most robust**.
Next, we analyze the right class confidence levels of the windows near those selected by the algorithm. As shown in Figure 8.12, while the desired window had the highest correct class confidence, neighboring windows often showed similar confidence levels. This suggests that even if the algorithm fails to identify the exact desired window from the streaming data, **in most cases, selecting neighboring windows provides a degree of flexibility in terms of accuracy.**

We now assess the effectiveness of SENTREC by comparing it with other statistical methods and random window selection (Table 8.2). To do this, we select a single window from the test set observations based on statistical feature benchmarks and in random fashion, and record the mean accuracy of these windows. Our results show that **SENTREC outperforms these methods in terms of accuracy** (Figure 8.13).

We now evaluate how effective the windows are under realistic interference conditions. We design a random noise and implement that on all the 50 sample sub-windows of an event. We analyze the impact of noise on the correct class confidence across the sub-windows. As observed in Figure 8.14, **SENTREC identifies robust windows that can withstand noise more effectively than others.**



(a) W:131,C: 1.0 **(b) W:114,C: 0.60**

(c) W:86,C: 1.0 **(d) W:76,C: 0.3875**

Figure 8.11: A few observations regarding confidences of the most robust windows [W: Window Number, C: Confidence for the right class] (x-axis: sample, y-axis: confidence). As seen, regarding confidence, robust windows are not always the best performer

8.9 Exploring methods to identify the target window from streaming data

As described in section 4.7, we first evaluate three different approaches using offline data. The idea is to select the best-performing approach for deployment in the sensor system to handle real-world streaming data. Since we focus on implementing occupant detection, *we are conducting the analysis exclusively on the PV dataset.*

Table 8.2: Different Methods Studied for Window Selection

Method	Acronym
Max Skewness	Skew
Max Variance	Variance
Max Difference [(Max val- Min val)]	Differ
Mean Absolute Deviation [abs(data - mean)]	Ratio
Max Slope	Slope
Random Selection of a window	Random
SENTREC Approach	Robust

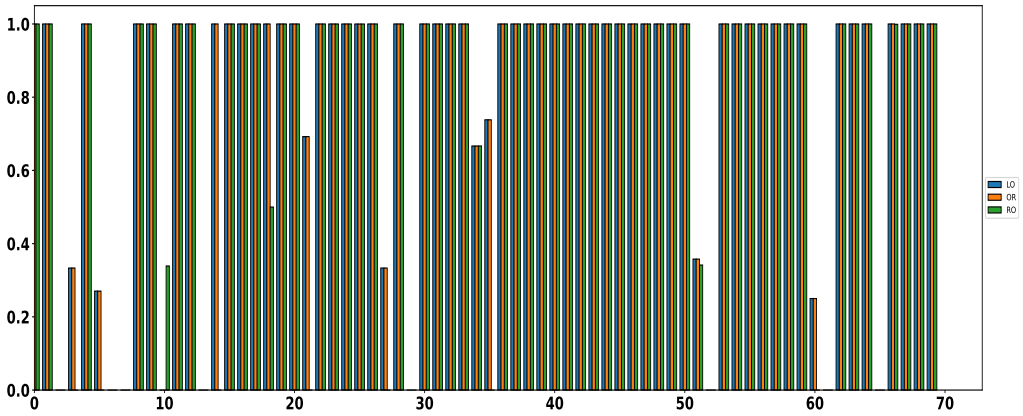


Figure 8.12: For successful attack observations: Confidence of the correct class for the most robust windows, along with neighboring windows offset by 3 (LO: Left Offset, RO: Right Offset, OR: Original Window) (x-axis: test example no. y-axis= correct class confidence). Despite similarities, robust windows outperform offset windows.

8.9.1 DTW based approach

After selecting the reference shape (Figure 8.15-a), we plot the distribution of the most similar window from an example (or in other words, record the minimum DTW distances: Figure 8.15-d) alongside the DTW distance cdf for the detected windows by our algorithm (Figure 8.15-b). As seen, **the minimum DTW distance window does not always represent the desired window**. To set an optimum value, we vary the DTW threshold over a range (Figure 8.15-e) and record the first window that satisfies the criterion as the most robust. After recording the desired window from each of the test set observation, we calculate the test set accuracy.

8.9.2 Threshold based approach

As shown in Figure 8.13, the statistical feature with accuracy closest to `SENTREC` is **Max. Difference (MAD)** *(the range between the maximum and minimum values within a window)*. For that, we assume that the MAD window from an observation likely represents the most robust window. We then record the MAD window from each test example. To set a threshold for the real-time data stream, we vary the threshold values, identify the first window that meets the threshold, and record the accuracy of the test set.

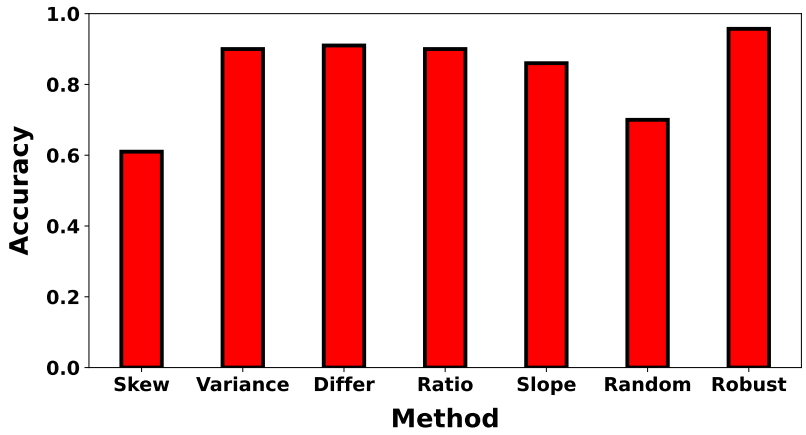


Figure 8.13: Comparing the accuracy of different methods on the test set against `SENTREC`

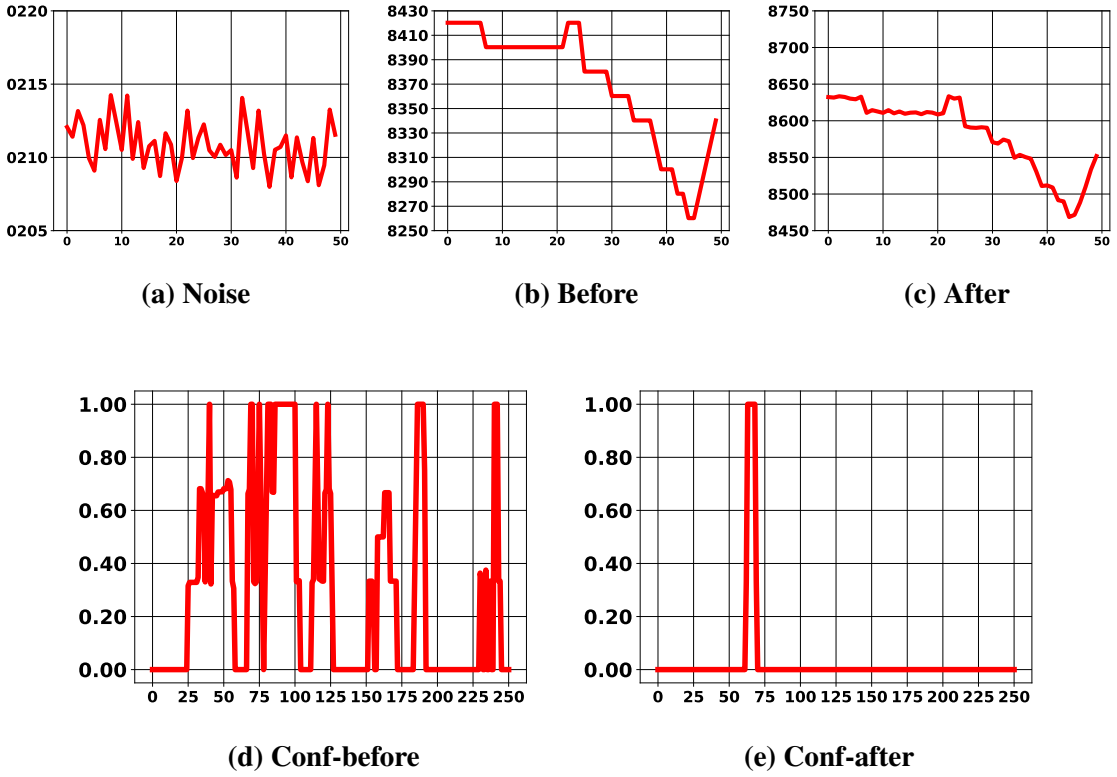
8.9.3 Machine Learning Approach

After gathering all the robust examples from the test set, we train three different ML-based classifiers to select the appropriate window from the data stream, after tuning the best parameters using *Gridsearch*. Based on the performance and resource-friendly implementation in edge devices (Figure 8.17 (top-left)), we choose **KNN**. Now we run the **KNN** classifier on the test set windows, record the first window that is classified as the robust one from each observation, and record the classification accuracy.

Figure 8.17 (top-right) exhibits the comparison among three approaches. We include AUC scores alongside accuracy to highlight the model's ability to distinguish among occupants. As seen, for the test set observations, **DTW outperforms the other two approaches**. As observed from the confusion matrix [Figure 8.17 (bottom-left)], there was no common pattern for the misclassifying examples for DTW. However, their performance in filtering non-targeted events remains to be analyzed, which is addressed in the evaluation section.

8.10 Implementation

Figure 8.18 illustrates the deployment layout in realistic testbed. The hardware details were the same as the original dataset literature (details in [95]). Apollo samples a PV cell data with a periodicity of 20ms and stores the samples in a buffer. Once every 2.5 seconds, the onboard KNN

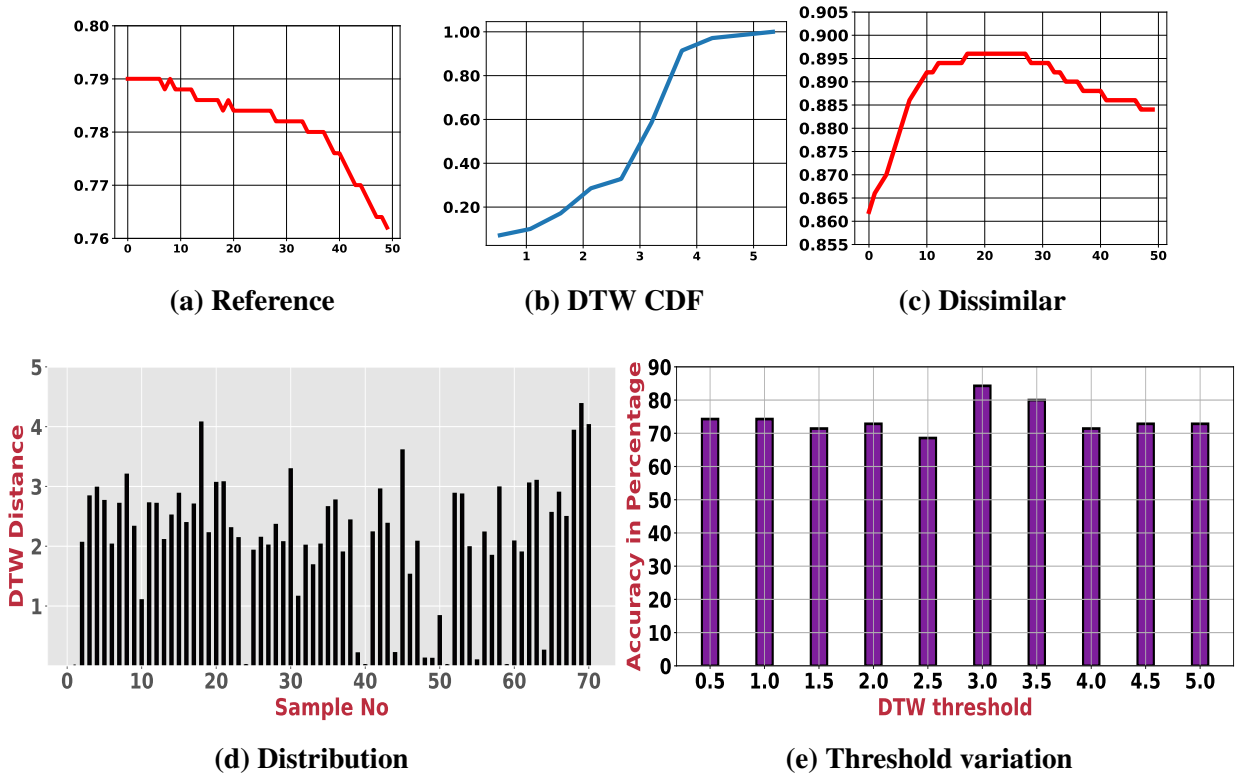


(a) Noise **(b) Before** **(c) After**

(d) Conf-before **(e) Conf-after**

Figure 8.14: (a) Introduction of random noise (a) to an observed example: (b) and (c) illustrate the impact of noise on the 50-sample window before and after the noise is applied, respectively (x-axis: time samples, y-axis: voltage values) while (d) and (e) depict the confidence in the correct class before and after the noise introduction (x-axis: window number, y-axis: correct class confidence). As observed, our reported robust window, number 68, demonstrates greater resilience to noise compared to the other windows.

classifier checks a window of the most recent 50 samples to determine if the shape of the input data matches training data traces to determine if a legitimate event has occurred. The periodicity of 2.5 seconds was chosen to accommodate the classification while ensuring an event can still be registered within the timespan of two sampling windows. If the classifier detects a legitimate event, Apollo applies a compression algorithm to reduce the transmission payload size. For our implementation, we quantize the float-type input data to reduce the precision with the benefit of better compression potential. Next, we apply delta encoding to minimize data redundancy. Finally, we utilize zlib's $deflate$ compression algorithm **zlib** to further reduce the payload size. We chose these algorithms to balance the tradeoff between precision and loss during compression. The size of the compressed output ultimately depends on the variability of the input signal. Once the final



(a) Reference **(b) DTW CDF** **(c) Dissimilar**

(d) Distribution **(e) Threshold variation**

Figure 8.15: (a) Reference window used for selecting the ideal window (time sample vs voltage), (b) DTW distances between the reference window and robust windows in the PV dataset (distance vs cdf), (c) The most dissimilar window identified (time sample vs voltage), (d) DTW distances for PV test set (e) Test set accuracy (in percentage) over different thresholds

compressed signal is obtained, Apollo transmits the data over a BLE advertisement to the server side for processing. We implement a buffer swap mechanism to prevent sample loss during the classification process because the classification and compression steps take the buffered PV cell samples as input.

8.11 Performances at Realistic Testbed

We evaluate our proposed architecture by performing the following experiments: (1) Targeted event: Multiple Exit/entry events of two individuals with different BMI. (2) Non-targeted event: Chair dragging/ Door Dragging through the entrance. For evaluation, we consider the events detected by

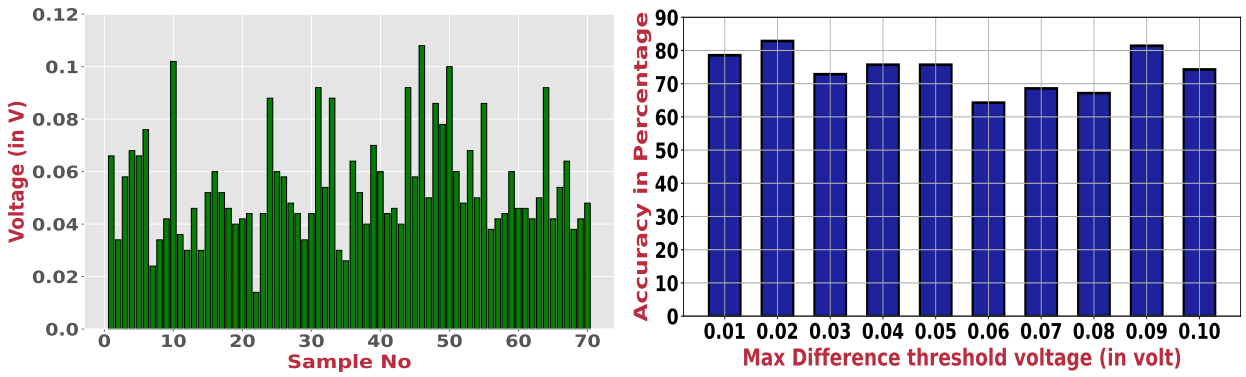


Figure 8.16: Distribution of max MAD from each example of the test set (left). Accuracy of the test set when max MAD threshold was varied

the PIR sensor and evaluate the metrics proposed by the three approaches. As observed, the **threshold-based approach has failed to filter the chair-dragging event.** Both the DTW and ML-based methods successfully filtered the non-targeting events (shown in Figure 8.19). However, given our objective to implement the entire architecture within constrained on-board resources, we opted for **the ML-based approach** to filter out non-target events and retain the desired segments for subsequent processing. As shown in Figure 8.19, a single window was chosen from an entry event in a realistic testbed setting. After compression and reconstruction, the occupant was correctly identified at the receiving end with the pre-trained KNN classifier.

After implementing the real-world deployment, we now calculate the data efficiency **based on the number of raw bytes compare to the full length observation**, both after segment selection and lossless compression. As observed in Figure 8.20, **98.79% of reduction of bytes** in data transmission compare to full-length observation was achieved through compression and reconstruction, while maintaining accurate occupant identification at the receiver using the pre-trained KNN classifier.

Now we calculate the average energy draw by the nrf51822 dk board at different stages with different payloads with Online Power Profiler for Bluetooth LE and represent a comparative analysis. As seen, the average current draw in advertising mode for the full length event has come down from 60.3μ**A** to only 0.77μ**A** (with a sampling frequency of 50 Hz), which is an improvement of around **98.72%**.

Now let us compare how our approach outperforms some earlier endeavors. As seen in Table 1.1, SENTREC utilizes existing simplistic sensor framework with ultra-low energy edge device for the

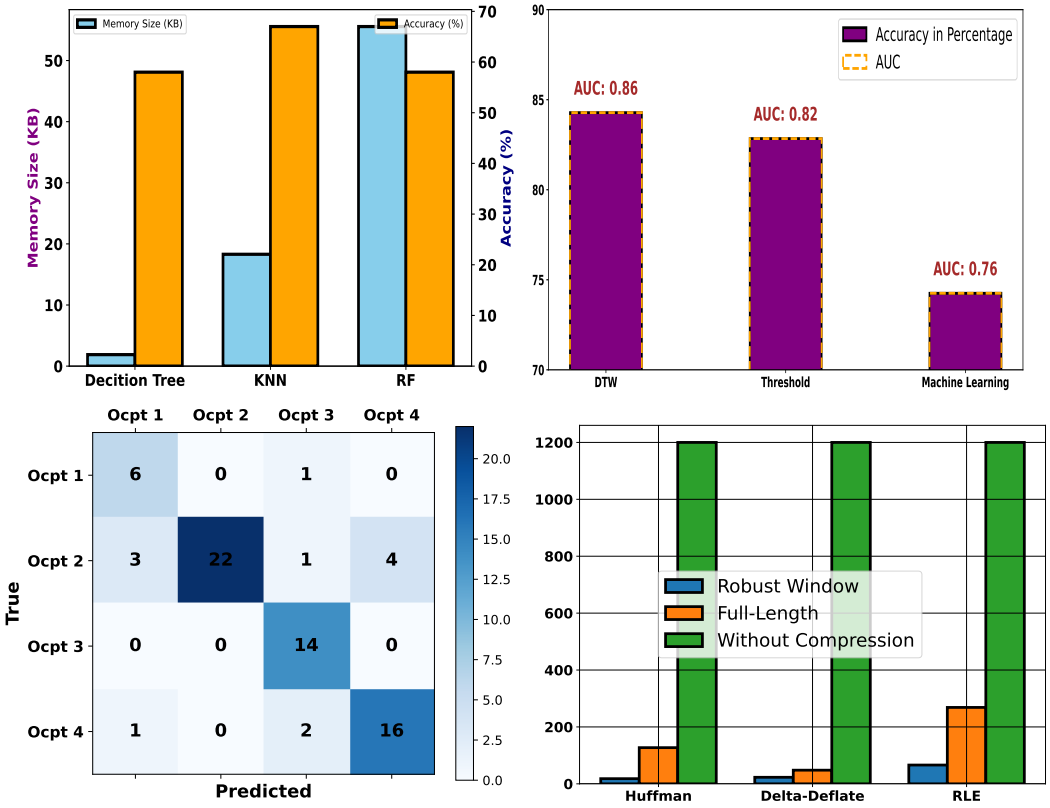


Figure 8.17: Test set accuracy and resource requirement comparison among different ML methods (top-left) and different approaches for desired window selection (top-right). Confusion matrix for DTW exhibits no common pattern of misclassification (bottom-left). The mean number of bytes calculated for the compressed robust window, compressed full-length observation, and uncompressed window for PV variation data used in occupant identification (bottom-right) [y-axis: number of bytes]. As observed, *Delta-Deflate method* outperforms RLE and Huffman method

task, at the same time, exhibiting elevated classification accuracy compare to the similar works.

8.12 Discussion

8.12.1 Latency

As outlined in the implementation, there is a 2.5-second delay associated with the selection of the window and the compression process. During this interval, if additional occupant movement occurs, the system may continue processing the previous activity, potentially overlooking the new movement.

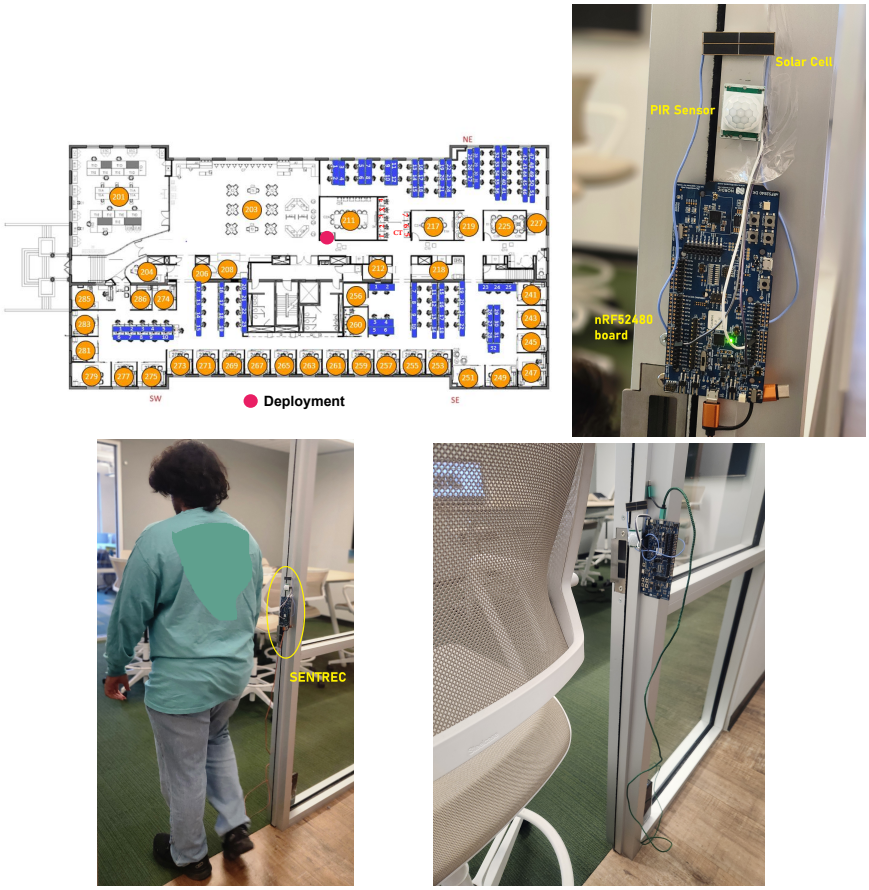


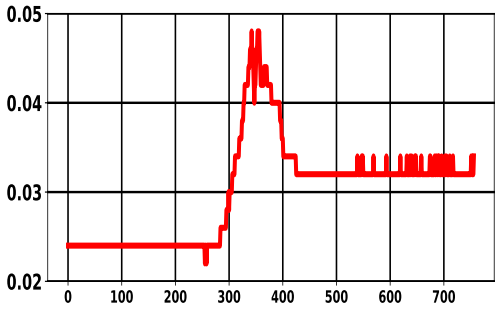
Figure 8.18: Floor plan depicting the placement of sensor installed on the door (top-left), Installed *Apollo* prototype (top-right), Targeted occupant movement event (bottom-left), A non-targeted chair dragging event (bottom-right)

8.12.2 Detection

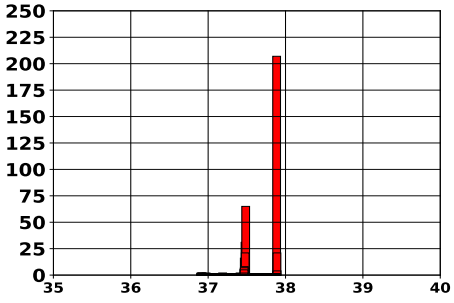
The approach relies on predefined patterns and may struggle to adapt to new occupants without retraining or updating the classifiers. Factors such as variations in walking speed, stride length, and changes in movement behavior due to health conditions or environmental constraints (e.g., crowded spaces or obstacles) could introduce inconsistencies in walking behavior that may affect accuracy. The amount of data/energy reduction will be determined by the frequency of occupant movement and the proportion of targeted to non-targeted events occurring within a given timeframe. For simplicity, we assume that one participant is causing the voltage variations and movements occur as independent events, without concurrent perturbations (like entrance and door dragging at the same time) that could disrupt the patterns.

System	Sensor Data	Utilizing existing sensors	Energy Intake/Real-World Noise Considered	Inference Platform	Processing Latency	Reported Accuracy
SenseTribute [102]	Accelerometer and Gyroscope (Five Pairs)	Yes	No	Arduino Uno	Not mentioned	74%
Non-Intrusive [103]	Temperature, humidity, pressure, light, motion, sound, and CO2 levels	No	No	NVIDIA Jetson Nano Developed Kit	1 min	99.75%
Ultrasound [104]	Three ultrasonic ping sensors	No	No	Raspberry PI 2 model B	Not mentioned	95%
Our System	**Single PV module**	**Yes**	**Yes**	nrf52480dK **(ultra-low power)**	**2.5s**	**100% (4 targeted, 2 non-targeted)**

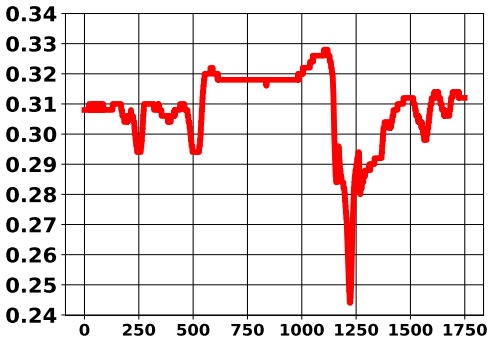
Table 8.3: A comparison of our approach with some similar works done for Occupant Identification



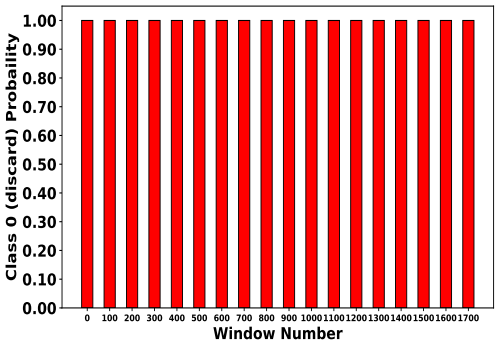
(a) Chair Dragging



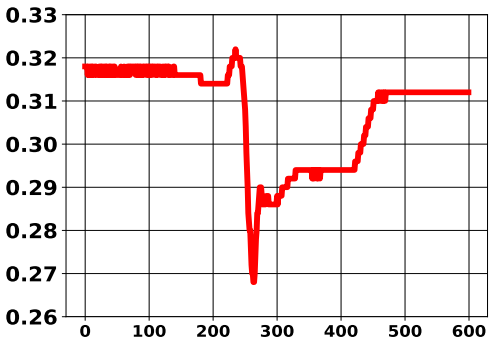
(b) DTW Dist. of Chair Dragging



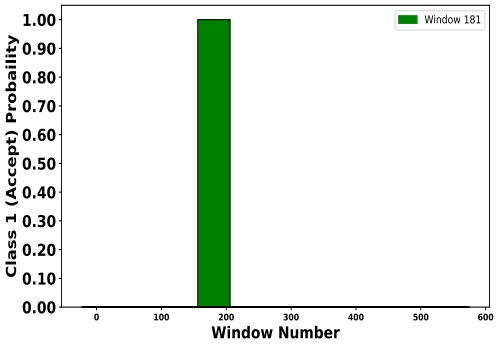
(c) Door Dragging



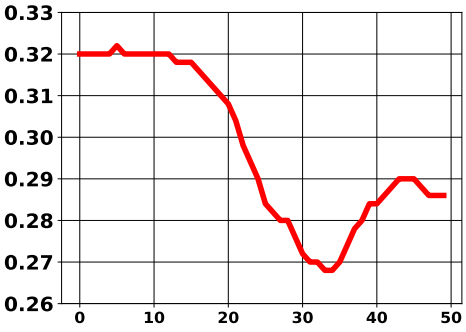
(d) Prob with KNN for different windows



(e) Entrance



(f) The first allowed window for classification



(g) Selected window

Figure 8.19: A non-targeted event involving chair dragging (a). DTW distances among 50 samples window of the chair dragging event and the reference window (x-axis:window number, y-axis: DTW distance). As seen the distances are much higher compare to targeted occupant detection windows shown in Figure 8.15. A non targeted door-dragging event (c) where all the windows were rejected by the KNN classifier (d), An entry event (e) where the first eligible window for classification was window number 181 (f). The allowed window (g).

Raw bytes (full)	Raw bytes (seg)	Comp. bytes (full)	Comp. bytes (seg)	Eff.(full)	Eff.(seg)
2400	200	112	29	**98.79%**	**85.5%**

Figure 8.20: Data efficiency evaluation for real world deployment (full: Full Length, seg: 50 samples Segment, Eff: Achieved Byte Efficiency)

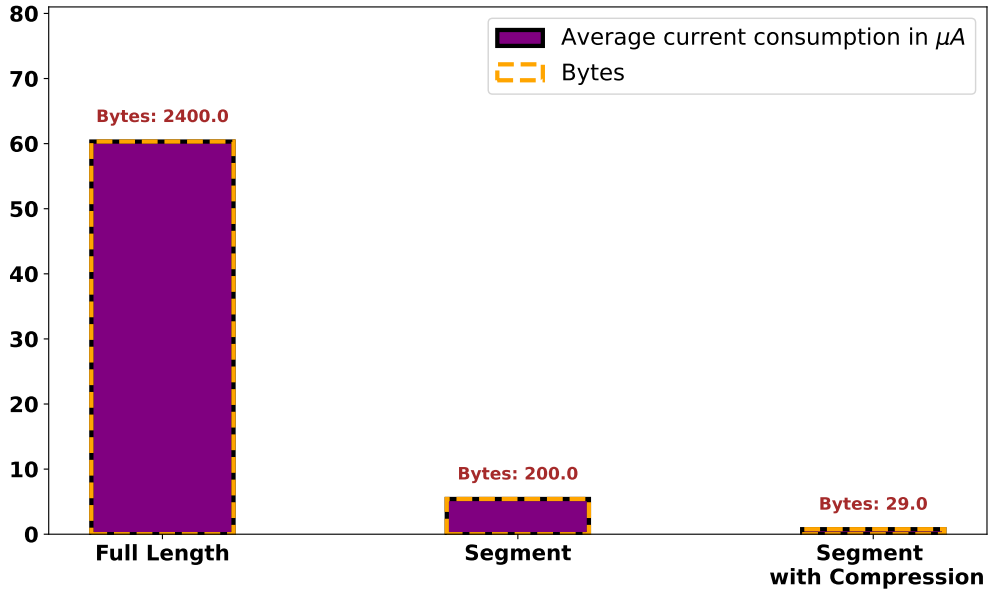


Figure 8.21: Comparison of average current draw in μA for different payloads. As seen, after segment selection and lossless compression, energy draw was reduced significantly

Chapter 9

CONCLUSION

Smart indoor light sensors are among the many smart devices integrated into our surroundings, aiming to enhance the quality of life, especially in indoor environments. Beyond promoting well-being, these specialized sensors are playing a crucial role in optimizing indoor lighting setups, managing agricultural environments, ensuring compliance with building-specific lighting guidelines, highlighting products in warehouses, accentuating artworks in galleries and museums and so on. These applications require energy-efficient sensors operating continuously and a reliable post-processing platform to fulfill their intended purpose. Unfortunately, the growing use of indoor light sensors in smart environments has not been matched by adequate focus on their efficient operation and reliable classification or event detection across diverse real-world scenarios. Current techniques result in sensors that consume excessive power and are data inefficient. When intended to identify the light source type in the surroundings, they are able to perform single source classification only at familiar environments. The classification accuracy drops after real life deployment, due to deployment parameter variations, irregular signal patterns from random movements or smart on/off scenarios, hindering their widespread adoption. Current techniques remain suboptimal for sensor-dense environments, where light sensors may encounter communication overhead, or under noisy conditions, where sensor readings can be influenced by interference from adjacent sensors.

With the number of IoT devices surpassing the global population and billions more expected in the next decade, the adoption of Green-IoT has become a clear necessity. It calls for exploring the potential of repurposing existing infrastructure, driving further exploration into novel applications of indoor light sensors beyond their intended function. Unfortunately, little effort has been made to explore how data from light sensors can be used beyond describing the surrounding lighting

environment and uncovering the additional information they may encode. Additionally, no studies have explored the privacy concerns that may arise in indoor environments when installed indiscriminately. Finally, despite being referred to as smart sensing systems, in many cases, they are not truly smart, as their current operational techniques are inefficient in terms of both data and energy usage.

In this dissertation, we design multiple strategies to address the aforementioned concerns. After collecting data from a diverse range of artificial *(Incandescent, LED and CFL)* and natural source, we discover that, in the real world, different types of sources share common light properties, making source identification a non-linear problem. We find that this complexity shuts the door for threshold-based Indoor Light Source Detection and opens the door for Machine Learning and Neural Network-based classification techniques. We learn that the classification accuracy can fluctuate based on the placement of sensors, the number of samples, and the classifier model. We find that a limited training dataset constrains the universal deployability of the system. However, augmenting the dataset with examples that simulate smart on-off scenarios, along with representations of unseen sources from the same distribution, significantly enhances accuracy and performance in real-world landscapes.

We observe that existing classifiers are currently are designed to identify single source environments, while modern lighting architectures may consist of multiple types of light sources. We also observe that when samples within a timeframe exhibit minimal variation, **Dimension Reduction** can be useful. This method effectively reduces on-air data, making classification feasible even in sensor-dense environments with packet loss. However, the most effective Dimension Reduction technique depends on the data characteristics, while its on-device deployability is influenced by the device's available resources.

We find that the typical setup of light sensors involves monitoring stable environments for extended periods, leading to unnecessary data accumulation and energy consumption on the sensor node, which reduces system efficiency. This can be improved by detecting environmental changes and only classifying the new environment. Additionally, we recognize that not all samples within a given timeframe are equally important for classification. Significant data efficiency can be achieved by identifying key observations within a specific timeframe and applying a lossless compression algorithm.

While utilizing RGB information from screen with typical light sensors, we observe that different activities exhibit unique patterns while also sharing common color properties, making activity identification a complex, non-linear problem. By positioning the sensor at various distances and angles, we found that the generated patterns remain unaffected by placement variations. However, influence from indoor light sources, activity transitions, and unfamiliar activities not included in the training set can significantly affect real-world classification performance. Retraining the classifier with enhanced dataset simulating room lighting conditions, switching events, and examples of unfamiliar screen settings can improve **Screen Activity Identification** in real-world. This study explores the potential of indoor light sensors as passive screen activity detectors while emphasizing the importance of strategic sensor placement to address privacy concerns in indoor settings (shown in `ScreenSense` framework).

When analyzing events from indoor PV sensors, we find that the events are encoded exclusively in short bursts within the extended sensor recordings. Additionally, sensors may be sensitive to multiple events for which classification was not intended. We found that selecting the most confident segment of a certain length did not result in the most robust one. A combination of robustness and confidence is necessary for segment selection. Significant data efficiency can be achieved for the necessary information used in post-processing classification by selecting the appropriate lossless compression technique. And finally, it is possible to include these techniques for filtering segment from real-world streaming data with low power embedded systems (shown in `SENTREC` framework)

Both in `ScreenSense` and `SENTREC` project we address the issues of privacy in different manners. In `ScreenSense` project, we have demonstrated that when an indoor light sensor is placed near the working station, it is capable of exploring the privacy sensitive screen activity of an user. As a result, collected color sensitive information should be dealt carefully so that it has access only to the specific person/authority of interest. In the `SENTREC` project, occupant detection is directly linked to indoor privacy, supporting tasks such as identifying intruders in restricted areas.

In multiple of our projects, **KNN** was reported as the best performing classifier, especially classifying the controlled observations. This implicates several key characteristics. At first, the data has low intrinsic complexity, meaning they are well separated within the feature space. We primarily work with primary dataset, where there were a few mislabeled or outlier examples. This also helped KNN to gain the upperhand, as it's performance detereorates with mislabled/noisy data.

Any finally, KNN does not perform well with data with linear separability or of very high dimension. This suggests RGB information from light sources/screen are locally smooth, and examples from the same class exhibits locality in lower-dimensional manifolds.

9.1 Limitations and Future Work

While this dissertation has proposed solution regarding real-world performance of classifiers and data efficient classification technique for long term monitoring, the presented approaches still have some limitations, which are worth exploring in the future. In this section, we discuss some limitations of our current approaches and outline potential avenues for future research, with the goal of enhancing the effectiveness and efficiency of real-world deployment of Indoor Light Sensors.

9.1.1 Indoor Light Source Classification

For daylong indoor light source classification, we focus on four main light types. However, in real-world scenarios, other types, such as *Halogen Bulbs*, may exist. Since halogen lights have similar RGB characteristics to incandescent lights, our trained classifier would identify them as incandescent. Increasing the number of classes allows for the identification of a wider range of indoor source types. During the control tests, we did not extensively test light bulbs with colored glass or rotating searchlights. The LPCSB board was designed for low-power operation, which limits its ability to perform on-board classification. In future, color sensing boards could be redesigned to support on-board classification and transmit only the results. Since dimming affects only the magnitude of color components while preserving temporal patterns, classification accuracy remains unaffected by dimming, given a static positioning of both the source and the sensor. Under the current settings, the LPCSB could not detect lighting scenarios with very low light levels. However, To address this, higher gain settings should be configured to capture extreme low-light conditions, although this would increase power consumption.

9.1.2 Multi-type source Classification

As shown in the analysis, we have dealt with only 10 different classes. When there are more than one type of artificial light present within the same category *(like 2 different types of CFLs)* or there

are multiple lights of the same type *((same 2 CFL bulbs)*, our classifier will detect the light that as the single class *(here, CFL)*.

9.1.3 Dimension Reduction for Data Efficient Classification

In our work on identifying multiple source types, the classifier is limited to detecting up to two types simultaneously and cannot identify three or more types at once. Expanding the number of classes could enhance detection of more diverse indoor lighting scenarios. Kernel-PCA proved to be the most effective dimensionality reduction technique, but it was also memory-intensive. Future research should focus on exploring non-linear dimensionality reduction methods that are lightweight and suitable for deployment on edge devices.

9.1.4 Daylong Light Exposure Analysis

In this work, once a sample is identified that meets the switch-over lighting criteria, the algorithm analyzes the next 25 samples for classification. However, if a switching event occurs within this period, the algorithm fails to detect it. For the classification task, we focused on lossless compression of RGB information. However, exploring lossy compression could offer an additional reduction in classification data size, which needs to be investigated.

9.1.5 Passive Sensing of On-screen Activities

For this study, we focused on identifying five specific screen activities. However, in real-world scenarios, screen activities are diverse, and distinguishing between them can be challenging. For instance, reading text on a video combines multiple activities, making categorization complex. Additionally, we did not account for scenarios where color-sensitive information from multiple displays affects sensor data equally. Future research should also explore methods for categorizing screen activities when multiple types of content are displayed simultaneously. Furthermore, future research should explore how the system can accurately determine a user's actual engagement in an activity, such as detecting if they are playing something on-screen but looking elsewhere or even sleeping.

9.1.6 Robust and Data Efficient Classification

Our goal was to implement the segment detection algorithm on a low-power edge device (e.g., nRF52840 DK). However, due to the limited resources of these boards, we were unable to use the DTW-based algorithm, even though it was the best performer for segment identification. For the performance metric, we use classification accuracy. However, in the future, other metrics could be considered, such as the number of events occurring within a specific timeframe and how many of the targeted ones were accurately detected. For power efficiency measurement, an alternative metric could be the number of advertisements required before and after adopting the technique.

REFERENCES

[1] R. L. Gregory, "Eye and brain: The psychology of seeing," 2015.

[2] T. Li, Q. Liu, and X. Zhou, "Practical human sensing in the light," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, 2016, pp. 71–84.

[3] E. Oberle, X. R. Ji, S. Kerai, M. Guhn, K. A. Schonert-Reichl, and A. M. Gadermann, "Screen time and extracurricular activities as risk and protective factors for mental health in adolescence: A population-level study," *Preventive Medicine*, vol. 141, p. 106 291, 2020.

[4] M. Ruger, M. C. Gordijn, D. G. Beersma, B. de Vries, and S. Daan, "Time-of-day-dependent effects of bright light exposure on human psychophysiology: Comparison of daytime and nighttime exposure," *American Journal of Physiology-regulatory, integrative and comparative physiology*, vol. 290, no. 5, R1413–R1420, 2006.

[5] L. Baandrup and P. J. Jennum, "Effect of a dynamic lighting intervention on circadian rest-activity disturbances in cognitively impaired, older adults living in a nursing home: A proof-of-concept study," *Neurobiology of Sleep and Circadian Rhythms*, vol. 11, p. 100 067, 2021.

[6] F. Ozkalayci, U. Kocabas, B. U. Altun, S. Pandi-Perumal, and A. Altun, "Relationship between melatonin and cardiovascular disease," *Cureus*, vol. 13, no. 1, 2021.

[7] *Sleep and sleep disorder statistics*, Jul. 2021.

[8] L. Tähkämö, T. Partonen, and A.-K. Pesonen, "Systematic review of light exposure impact on human circadian rhythm," *Chronobiology international*, vol. 36, no. 2, pp. 151–170, 2019.

[9] C. Ticleanu, "Impacts of home lighting on human health," *Lighting Research & Technology*, vol. 53, no. 5, pp. 453–475, 2021.

[10] N. Shishegar, M. Boubekri, E. A. Stine-Morrow, and W. A. Rogers, "Tuning environmental lighting improves objective and subjective sleep quality in older adults," *Building and Environment*, vol. 204, p. 108 096, 2021.

[11] K. Missildine, "Sleep and the sleep environment of older adults in acute care settings," *Journal of gerontological nursing*, vol. 34, no. 6, pp. 15–21, 2008.

[12] M. M. Ohayon and C. Milesi, "Artificial outdoor nighttime lights associate with altered sleep behavior in the american general population," *Sleep*, vol. 39, no. 6, pp. 1311–1320, 2016.

[13] V. Singhvi, A. Krause, C. Guestrin, J. H. Garrett Jr, and H. S. Matthews, "Intelligent light control using sensor networks," in *Proceedings of the 3rd international conference on Embedded networked sensor systems*, 2005, pp. 218–229.

[14] S. Nižetić, P. Šolić, D. L.-I. Gonzalez-De, L. Patrono, *et al.*, "Internet of things (iot): Opportunities, issues and challenges towards a smart and sustainable future," *Journal of cleaner production*, vol. 274, p. 122 877, 2020.

[15] X. Liu and N. Ansari, "Toward green iot: Energy solutions and key challenges," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 104–110, 2019.

[16] MMR, "Light sensors market – global industry analysis and forecast (2024-2030)," www.maximizemarketresearch.com, Tech. Rep. 2893, 2024.

[17] F. Murano *et al.*, "Lighting artworks in art exibitions," 2016.

[18] D. C. J. Neo, M. M. X. Ong, Y. Y. Lee, *et al.*, "Shaping and tuning lighting conditions in controlled environment agriculture: A review," *ACS Agricultural Science & Technology*, vol. 2, no. 1, pp. 3–16, 2022.

[19] P. Yeh, N. Yeh, C.-H. Lee, and T.-J. Ding, "Applications of leds in optical sensors and chemical sensing device for detection of biochemicals, heavy metals, and environmental nutrients," *Renewable and Sustainable Energy Reviews*, vol. 75, pp. 461–468, 2017.

[20] T. A. Nguyen and M. Aiello, "Energy intelligent buildings based on user activity: A survey," *Energy and buildings*, vol. 56, pp. 244–257, 2013.

[21] X. Wang, Y. Li, and H. Fan, "The associations between screen time-based sedentary behavior and depression: A systematic review and meta-analysis," *BMC public health*, vol. 19, pp. 1–9, 2019.

[22] P. Luo, D. Peng, Y. Wang, and X. Zheng, "Review of solar energy harvesting for iot applications," in *2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, IEEE, 2018, pp. 512–515.

[23] H. A. Krishna, N. Misra, and M. Suresh, "Solar cell as a capacitive temperature sensor," *IEEE transactions on aerospace and electronic systems*, vol. 47, no. 2, pp. 782–789, 2011.

[24] M. F. R. M. Billah, N. Saoda, V. A. L. Sobral, T. Routh, W. Wang, and B. Campbell, "Solarwalk: Smart home occupant identification using unobtrusive indoor photovoltaic harvesters," in *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, 2022, pp. 178–187.

[25] M. A. R. Ahad, A. D. Antar, and M. Ahmed, "Iot sensor-based activity recognition," *IoT Sensor-Based Activity Recognition*, vol. 2, 2020.

[26] A. Rayes and S. Salam, "The things in iot: Sensors and actuators," in *Internet of Things From Hype to Reality: The Road to Digitization*, Springer, 2022, pp. 63–82.

[27] R. Krishnamurthi, A. Kumar, D. Gopinathan, A. Nayyar, and B. Qureshi, "An overview of iot sensor data processing, fusion, and analysis techniques," *Sensors*, vol. 20, no. 21, p. 6076, 2020.

[28] T. M. Brown, G. C. Brainard, C. Cajochen, *et al.*, "Recommendations for daytime, evening, and nighttime indoor light exposure to best support physiology, sleep, and wakefulness in healthy adults," *PLoS biology*, vol. 20, no. 3, e3001571, 2022.

[29] NIH, "Vitamin d fact sheet for health professionals," National Institute of Health, Tech. Rep., 2021.

[30] X. Li, B. Hou, R. Zhang, and Y. Liu, "A review of rgb image-based internet of things in smart agriculture," *IEEE Sensors Journal*, 2023.

[31] X. Ma, S. Bader, and B. Oelmann, "Characterization of indoor light conditions by light source classification," *IEEE Sensors Journal*, vol. 17, no. 12, pp. 3884–3891, 2017.

[32] B. Politi, A. Foucaran, and N. Camara, "Low-cost sensors for indoor pv energy harvesting estimation based on machine learning," *Energies*, vol. 15, no. 3, p. 1144, 2022.

[33] T. Martire, P. Nazemzadeh, A. Sanna, and D. Trojaniello, "Digital screen detection enabled by wearable sensors: Application in adl settings," in *2018 International Conference on Intelligent Systems (IS)*, IEEE, 2018, pp. 584–588.

[34] A. electronics, "Solar cells-slmd121h04l," https://www.allicdata.com/products/slmd121h04l/2808328.html, Tech. Rep., 2024.

[35] A. Panahi, "The health risks associated with energy efficient fluorescent, leds, and artificial lighting," in *Photonics Applications for Aviation, Aerospace, Commercial, and Harsh Environments V*, International Society for Optics and Photonics, vol. 9202, 2014, 92020K.

[36] K. M. Zielinska-Dabkowska, *Make lighting healthier*, 2018.

[37] A. Amirazar, M. Azarbayjani, M. Molavi, and M. Karami, "A low-cost and portable device for measuring spectrum of light source as a stimulus for the human's circadian system," *Energy and Buildings*, vol. 252, p. 111 386, 2021.

[38] A. Fernández-Montes, L. Gonzalez-Abril, J. A. Ortega, and F. V. Morente, "A study on saving energy in artificial lighting by making smart use of wireless sensor networks and actuators," *IEEE network*, vol. 23, no. 6, pp. 16–20, 2009.

[39] A. Sarris, "LPCSB: A Device to Distinguish Between Natural and Artificial Light," M.S. thesis, University of Virginia, Charlottesville, Virginia, 2020.

[40] J. Yoon, D. Jarrett, and M. Van der Schaar, "Time-series generative adversarial networks," *Advances in neural information processing systems*, vol. 32, 2019.

[41] Q. Xu, G. Huang, Y. Yuan, *et al.*, "An empirical study on evaluation metrics of generative adversarial networks," *arXiv preprint arXiv:1806.07755*, 2018.

[42] K. Shmelkov, C. Schmid, and K. Alahari, "How good is my gan?" In *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 213–229.

[43] A. Figueira and B. Vaz, "Survey on synthetic data generation, evaluation methods and gans," *Mathematics*, vol. 10, no. 15, p. 2733, 2022.

[44] Z. Yang, Y. Li, and G. Zhou, "Ts-gan: Time-series gan for sensor-based health data augmentation," *ACM Transactions on Computing for Healthcare*, vol. 4, no. 2, pp. 1–21, 2023.

[45] C. Esteban, S. L. Hyland, and G. Rätsch, "Real-valued (medical) time series generation with recurrent conditional gans," *arXiv preprint arXiv:1706.02633*, 2017.

[46] X. Lyu, H. Tian, L. Jiang, *et al.*, "Selective offloading in mobile edge computing for the green internet of things," *IEEE network*, vol. 32, no. 1, pp. 54–60, 2018.

[47] A. Longo, M. Zappatore, M. Bochicchio, and S. B. Navathe, "Crowd-sourced data collection for urban monitoring via mobile sensors," *ACM Transactions on Internet Technology (TOIT)*, vol. 18, no. 1, pp. 1–21, 2017.

[48] L. Van Der Maaten, E. O. Postma, H. J. Van Den Herik, *et al.*, "Dimensionality reduction: A comparative review," *Journal of machine learning research*, vol. 10, no. 66-71, p. 13, 2009.

[49] M. Ghashami, D. J. Perry, and J. Phillips, "Streaming kernel principal component analysis," in *Artificial intelligence and statistics*, PMLR, 2016, pp. 1365–1374.

[50] A. Shawqi Jaber and A. Kadhum Idrees, "Adaptive rate energy-saving data collecting technique for health monitoring in wireless body sensor networks," *International Journal of Communication Systems*, vol. 33, no. 17, e4589, 2020.

[51] T. Sheltami, M. Musaddiq, and E. Shakshuki, "Data compression techniques in wireless sensor networks," *Future Generation Computer Systems*, vol. 64, pp. 151–162, 2016.

[52] M. Ojala and G. C. Garriga, "Permutation tests for studying classifier performance.," *Journal of machine learning research*, vol. 11, no. 6, 2010.

[53] Y. Zhang, *On the importance of light source classification in indoor light energy harvesting*, 2018.

[54] C. Wilkinson, F. Low, and P. Gluckman, "Screen time: The effects on children's emotional, social, and cognitive development," 2021.

[55] H. Liu, "Various types of screen time and their influences on adolescents' depression,"

[56] E. Neophytou, L. A. Manwell, and R. Eikelboom, "Effects of excessive screen time on neurodevelopment, learning, memory, mental health, and neurodegeneration: A scoping

review," *International Journal of Mental Health and Addiction*, vol. 19, no. 3, pp. 724–744, 2021.

[57] H. C. Woods and H. Scott, "# sleepyteens: Social media use in adolescence is associated with poor sleep quality, anxiety, depression and low self-esteem," *Journal of adolescence*, vol. 51, pp. 41–49, 2016.

[58] G. Lissak, "Adverse physiological and psychological effects of screen time on children and adolescents: Literature review and case study," *Environmental research*, vol. 164, pp. 149–157, 2018.

[59] L. K. Kaye, A. Orben, D. A. Ellis, S. C. Hunter, and S. Houghton, "The conceptual and methodological mayhem of "screen time"," *International Journal of Environmental Research and Public Health*, vol. 17, no. 10, p. 3661, 2020.

[60] G. Hisler, J. M. Twenge, and Z. Krizan, "Associations between screen time and short sleep duration among adolescents varies by media type: Evidence from a cohort study," *Sleep medicine*, vol. 66, pp. 92–102, 2020.

[61] S. K. Ernala, M. Burke, A. Leavitt, and N. B. Ellison, "How well do people report time spent on facebook? an evaluation of established survey questions with recommendations," in *Proceedings of the 2020 CHI conference on human factors in computing systems*, 2020, pp. 1–14.

[62] *Activtrak*, `https://www.activtrak.com/`, 2023.

[63] *Teramind*, `https://www.teramind.co/`, 2023.

[64] A. D. Miyazaki, "Online privacy and the disclosure of cookie use: Effects on consumer trust and anticipated patronage," *Journal of Public Policy & Marketing*, vol. 27, no. 1, pp. 19–33, 2008.

[65] C. Min, E. Lee, S. Park, and S. Kang, "Tiger: Wearable glasses for the 20-20-20 rule to alleviate computer vision syndrome," in *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*, 2019, pp. 1–11.

[66] T. Vuong, G. Jacucci, and T. Ruotsalo, "Watching inside the screen: Digital activity monitoring for task recognition and proactive information retrieval," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, pp. 1–23, 2017.

[67] B. Reeves, T. Robinson, and N. Ram, *Time for the human screenome project*, 2020.

[68] D. Mistry, M. Mridha, M. Safran, S. Alfarhood, A. K. Saha, and D. Che, "Privacy-preserving on-screen activity tracking and classification in e-learning using federated learning," *IEEE Access*, 2023.

[69] 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*, 2016, pp. 47–52.

[70] S. W. Cain, E. M. McGlashan, P. Vidafar, *et al.*, "Evening home lighting adversely impacts the circadian system and sleep," *Scientific reports*, vol. 10, no. 1, pp. 1–10, 2020.

[71] F. Wahl, J. Kasbauer, and O. Amft, "Computer screen use detection using smart eyeglasses," *Frontiers in ICT*, vol. 4, p. 8, 2017.

[72] T. Blascheck, A. Bezerianos, L. Besançon, B. Lee, and P. Isenberg, "Preparing for perceptual studies: Position and orientation of wrist-worn smartwatches for reading tasks," in *Proceedings of the Workshop on Data Visualization on Mobile Devices held at ACM CHI*, 2018.

[73] G. Lindgaard, G. Fernandes, C. Dudek, and J. Brown, "Attention web designers: You have 50 milliseconds to make a good first impression!" *Behaviour & information technology*, vol. 25, no. 2, pp. 115–126, 2006.

[74] R. R. Fletcher, D. Chamberlain, D. Richman, N. Oreskovic, and E. Taveras, "Wearable sensor and algorithm for automated measurement of screen time," in *2016 IEEE Wireless Health (WH)*, IEEE, 2016, pp. 1–8.

[75] B. J. Ferdosi, M. Sadi, N. Hasan, and M. A. Rahman, "Tracking digital device utilization from screenshot analysis using deep learning," in *Proceedings of International Conference on Data Science and Applications: ICDSA 2022, Volume 1*, Springer, 2023, pp. 661–670.

[76] N. Ram, X. Yang, M.-J. Cho, *et al.*, "Screenomics: A new approach for observing and studying individuals' digital lives," *Journal of adolescent research*, vol. 35, no. 1, pp. 16–50, 2020.

[77] L. Schwittmann, V. Matkovic, T. Weis, *et al.*, "Video recognition using ambient light sensors," in *2016 IEEE international conference on pervasive computing and communications (PerCom)*, IEEE, 2016, pp. 1–9.

[78] D. Avrahami, E. van Everdingen, and J. Marlow, "Supporting multitasking in video conferencing using gaze tracking and on-screen activity detection," in *Proceedings of the 21st International Conference on Intelligent User Interfaces*, 2016, pp. 130–134.

[79] G. De Palma, E. Sala, S. Rubino, *et al.*, "Objective evaluation of active interactions between the operator and display screen equipment using an innovative acquisition system," *Bioengineering*, vol. 10, no. 6, p. 686, 2023.

[80] J. Bouclé, D. Ribeiro Dos Santos, and A. Julien-Vergonjanne, "Doing more with ambient light: Harvesting indoor energy and data using emerging solar cells," in *Solar*, MDPI, vol. 3, 2023, pp. 161–183.

[81] V. M. Research, *Indoor Solar Cell Market Size And Forecast*, `https://www.verifiedmarketresearch.com/product/indoor-solar-cell-market//`, 2023.

[82] A. Chakraborty, G. Lucarelli, J. Xu, *et al.*, "Photovoltaics for indoor energy harvesting," *Nano Energy*, p. 109 932, 2024.

[83] N. Saoda, "Designing batteryless energy-harvesting sensors for sustainable internet-of-things," Available at `https://libraetd.lib.virginia.edu/public_view/tb09j716v`, PhD thesis, University of Virginia, Charlottesville, VA, Aug. 2023.

[84] M. Hanlon and R. Anderson, "Real-time gait event detection using wearable sensors," *Gait & posture*, vol. 30, no. 4, pp. 523–527, 2009.

[85] Z. H. Janjua, M. Vecchio, M. Antonini, and F. Antonelli, "Irese: An intelligent rare-event detection system using unsupervised learning on the iot edge," *Engineering Applications of Artificial Intelligence*, vol. 84, pp. 41–50, 2019.

[86] Q. Xu, Y. Chen, B. Wang, and K. R. Liu, "Trieds: Wireless events detection through the wall," *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 723–735, 2017.

[87] Y. Luo and S. Nirjon, "Spoton: Just-in-time active event detection on energy autonomous sensing systems," *Brief Presentations Proceedings (RTAS 2019)*, vol. 9, 2019.

[88] S. M. Errapotu, J. Wang, Y. Gong, J.-H. Cho, M. Pan, and Z. Han, "Safe: Secure appliance scheduling for flexible and efficient energy consumption for smart home iot," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4380–4391, 2018.

[89] P. F. Karjou, S. K. Saryazdi, P. Stoffel, and D. Müller, "Practical design and implementation of iot-based occupancy monitoring systems for office buildings: A case study," *Energy and Buildings*, p. 114 852, 2024.

[90] B. Kim and S. Lee, "On-nas: On-device neural architecture search on memory-constrained intelligent embedded systems," in *Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems*, 2023, pp. 152–166.

[91] M. A. Jan, F. Khan, S. Mastorakis, M. Adil, A. Akbar, and N. Stergiou, "Lightiot: Lightweight and secure communication for energy-efficient iot in health informatics," *IEEE transactions on green communications and networking*, vol. 5, no. 3, pp. 1202–1211, 2021.

[92] A. M. Hussein, A. K. Idrees, and R. Couturier, "Distributed energy-efficient data reduction approach based on prediction and compression to reduce data transmission in iot networks," *International Journal of Communication Systems*, vol. 35, no. 15, e5282, 2022.

[93] M. J. Rubin, M. B. Wakin, and T. Camp, "Lossy compression for wireless seismic data acquisition," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 1, pp. 236–252, 2015.

[94] H. M. Al-Kadhim and H. S. Al-Raweshidy, "Energy efficient data compression in cloud based iot," *IEEE Sensors Journal*, vol. 21, no. 10, pp. 12 212–12 219, 2021.

[95] N. Saoda, M. F. R. M. Billah, V. A. L. Sobral, and B. Campbell, "Solarwalk dataset: Occupant identification using indoor photovoltaic harvester output voltage," pp. 1031–1034, 2022.

[96] F. Karim, S. Majumdar, and H. Darabi, "Adversarial attacks on time series," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3309–3320, 2020.

[97] Z. Zhao, G. Chen, J. Wang, Y. Yang, F. Song, and J. Sun, "Attack as defense: Characterizing adversarial examples using robustness," in *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2021, pp. 42–55.

[98] M. S. Ayas, S. Ayas, and S. M. Djouadi, "Projected gradient descent adversarial attack and its defense on a fault diagnosis system," in *2022 45th International Conference on Telecommunications and Signal Processing (TSP)*, IEEE, 2022, pp. 36–39.

[99] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*, Ieee, 2017, pp. 39–57.

[100] B. Feldsar, R. Mayer, and A. Rauber, "Detecting adversarial examples using surrogate models," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1796–1825, 2023.

[101] N. Gong, Z. Wang, S. Chen, G. Liu, and S. Xin, "Decision boundary extraction of classifiers," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1651, 2020, p. 012 031.

[102] J. Han, S. Pan, M. K. Sinha, H. Y. Noh, P. Zhang, and P. Tague, "Smart home occupant identification via sensor fusion across on-object devices," *ACM Transactions on Sensor Networks (TOSN)*, vol. 14, no. 3-4, pp. 1–22, 2018.

[103] A. N. Sayed, F. Bensaali, Y. Himeur, and M. Houchati, "Edge-based real-time occupancy detection system through a non-intrusive sensing system," *Energies*, vol. 16, no. 5, p. 2388, 2023.

[104] N. Khalil, D. Benhaddou, O. Gnawali, and J. Subhlok, "Nonintrusive ultrasonic-based occupant identification for energy efficient smart building applications," *Applied Energy*, vol. 220, pp. 814–828, 2018.