SeniorSentry: Safeguarding AgeTech Devices and Sensors Using Contextual Anomaly Detection and Supervised Machine Learning

by

Achyuth Nandikotkur
B.Tech., Manipal Institute of Technology, 2017

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

SeniorSentry: Safeguarding AgeTech Devices and Sensors Using Contextual Anomaly Detection and Supervised Machine Learning

by

Achyuth Nandikotkur
B.Tech., Manipal Institute of Technology, 2017

Supervisory Committee

Dr. Issa Traore, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Mihai Sima, Department Member
(Department of Electrical and Computer Engineering)

Supervisory Committee

Dr. Issa Traore, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Mihai Sima, Department Member
(Department of Electrical and Computer Engineering)

ABSTRACT

With the ever-growing reliance on IoT-enabled sensors to age in place, a need arises to protect them from malicious activities by detecting attacks or other anomalies. In this work, we first confirm the presence of correlations between co-located sensors by statistically analyzing two public smart-home datasets and a dataset we collected from our lab. Then, we leverage the sliding window approach and supervised machine learning to develop a novel contextual-anomaly-detection model that reaches a true positive rate of 89.47% and a false positive rate of 0%. Furthermore, as homes become smarter with these IoT sensors, the underlying communication technology they employ becomes a target for attackers. Typically, these sensors are paired with a micro-controller that has an inbuilt communication module (e.g., Bluetooth/WiFi), to form an edge device that facilitates communication. Monitoring vitals, climate control, illumination control, fall detection, incontinence detection, pill dispensing, and several other functions are successfully addressed by these devices. The family of vulnerabilities recently found in the the Link Manager Protocol (LMP) and baseband layers of the Bluetooth Classic (BT Classic) stack called BrakTooth, poses a genuine threat to the availability of such devices. In response, our research introduces a cost-effective experimental active sniffer that captures traffic at both these layers of the BT Classic stack and utilizes supervised machine learning to detect Braktooth-based attacks.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my supervisor, Dr. Issa Traore, for his unwavering patience and invaluable guidance throughout this project. His support, encouragement, and mentorship have been instrumental in my journey, and I am truly grateful for his expertise and dedication to scientific research.

I am also profoundly thankful to Dr. Mohammad Mamun for his crucial support at significant junctures of this endeavor. His insights and assistance have greatly contributed to the success of this work. Furthermore, I extend my sincere appreciation to the lecturers who have provided me with the foundational knowledge and skills necessary to undertake this research. Their teachings and expertise have been instrumental in shaping my understanding of the subject matter.

To my parents, Dipak Kumar and Dr. Padmavathi, and my dear brother, Anirudh, for their unwavering support and love; and in loving memory of Lucy and Leo, my cherished companions.

Chapter 1

Introduction

1.1 Context

1.1.1 What is aging in place?

Aging in place refers to the idea of older adults maintaining their independence as they age by living in their own homes and communities without moving to a different living environment, such as an assisted living facility or a nursing home. According to World Population Prospects 2022 [1], the portion of the global population over 65 years of age is expected to grow from 10% in 2022 to 15% in 2050. This population growth at older ages is attributed to the increased life expectancy [2] and a sustained decline in fertility. As a result, the United Nations suggested that countries with aging populations must establish universal and long-term health care to prevent overwhelming health infrastructure. In response to this suggestion, many countries are increasingly turning to technology.

1.1.2 How can technology support aging in place?

The Internet of Things technologies can provide useful systems and solutions [3] that monitor personal health, provide medication reminders, dispense pills, curate mobility plans and gamify activity [4] so seniors can enjoy longer periods of healthy and functionally independent life, thus reducing the pressure on the health infrastructure. Furthermore, technology can also be useful in improving the quality of life of older adults having chronic conditions [5, 6]. Typically the following areas are monitored using technology to support aging in place.

- Physiological markers: Monitoring and analyzing blood pressure, heart rate, skin temperature, water intake, glucose levels, urine volume, body weight, and eye pressure can be helpful in evaluating cognitive performance, hydration, hypo and hyperglycemia, fatigue, heat stress, and more [7]. With this knowledge, corrective actions, such as controlling ambient temperature, providing medication reminders, dispensing pills, and suggesting physical activity, can be automatically taken to maintain these markers within the normal range.

- Activity: Monitoring an older adult's activity through the number of steps taken, time spent on exercise equipment, patterns of movement in the house, and the time spent outside can be beneficial in gauging mental and physical well-being and establishing a baseline trend. Any deviations from this baseline can provide a valuable context for care providers to take appropriate corrective actions.

- Socialization: Social isolation is linked to serious health conditions. It is associated with an 50% increased risk of dementia and a 29% increased risk of heart disease [8]. Therefore, monitoring the social interaction of an older adult and notifying family members of an anomalous drop in social interaction can be very useful in improving the quality of life.

Table 1.1, provides several examples of common devices that can be found in AgeTech smart homes.

1.1.3 Threats to AgeTech

Due to limited technical literacy and experience, as well as declining physical and mental abilities, older adults can be less aware and more susceptible to privacy and security risks [9]. As a result, the security challenges and attack vectors present in traditional IoT systems easily transfer to the systems used for aging in place. The traditional Internet of Things is typically composed of three basic layers: the perception layer, the network layer, and the application layer

- Perception Layer: The perception layer is the physical layer of the IoT architecture. It encompasses sensors, and actuators, called *things* and their interfaces to the environment. The major functions of this layer involve producing a numerical (continuous or discrete) representation of external conditions and transmitting it to higher layers through it's interfaces [10].

IoT Device	Protocol	Role in AgeTech	Potential Implementation
IP Cam	WiFi	Surveillance	ESP32-CAM with ESP32 MC
Thermostat	WiFi/Bluetooth	Temperature control	DHT22 with ESP32 MC
Flame Detector	WiFi/Bluetooth	Fire protection	MQ135, KY-026, with ESP32 MC.
Smart light	WiFi/Bluetooth	Illumination	BH1750 with ESP32 MC
Smart Lock	WiFi/Bluetooth	Security	ZFM-20, Servo motor, with ESP32 MC
SPO2, Heart rate monitor	Bluetooth	Monitoring vitals	MAX30101 with ESP32 MC
Fall Detection Sensors	Bluetooth, Others	Fall detection	MEMS - MPU6050 with ESP32 MC
Smart Pillbox	WiFi/Bluetooth	Medicine administration	Chassis, Servo motors and ESP32 MC
Smart Scale	WiFi/Bluetooth	Weight measurement	FC2231, HX711 with ESP32
Smart Diaper	Bluetooth	Incontinence measurement	SEN-13322 with ESP32 MC

Table 1.1: Examples of commonly used AgeTech devices.

- Network Layer: The network layer is primarily responsible for interconnecting the various *things* in the perception layer and establishing their connection to a server or an edge node. Communication standards like Wi-Fi, Bluetooth, and ZigBee form a part of this layer. The use of wirless technologies is instrumental in this layer to allow for deployment in critical environments with minimal human effort required for monitoring, and maintenance.

- Application layer: The application layer sits at the top of the IoT architecture and is responsible for processing and analyzing the data collected from the perception layer and presenting it to the users. It further enables users and applications to interact with *things* of the perception layer in a secure manner.

Table 1.2 summarizes the various threats at these different layers along with the traditional counter measures to mitigate them. Yet, these countermeasures might not be sufficient on their own; a defense-in-depth approach is essential to further enhance security.

Layer	Types of Attacks	Countermeasures
Perception	Physical tampering	Implement physical security measures
	Sensor spoofing	Use tamper-evident packaging for devices
	Side-channel attacks	Employ strong authentication for device access
	Firmware or software vulnerabilities	Regularly update firmware and software
Network	Man-in-the-Middle (MitM) attacks	Encrypt communication between devices
	Denial-of-Service (DoS) attacks	Use secure protocols for data transfer
	Network scanning and reconnaissance	Implement access control and authentication
	Network eavesdropping	Monitor network traffic for anomalies
Application	Application-level vulnerabilities	Apply secure coding practices
	API attacks	Regularly update and patch applications
	Command injection	Implement secure APIs and data validation
	Data breaches	Use encryption for sensitive data

Table 1.2: Types of Attacks in IoT Architecture Layers

1.2 Research Problem

In IoT smart homes, sensors are used at the perception layer to interface with the environment and gather data representative of the ambient conditions. The time series data produced by these sensors is typically sent to a base station or a fog node, using wireless communication technologies like Bluetooth or WiFi. This is usually accomplished by integrating the sensor with a microcontroller that has built-in WiFi/Bluetooth support (e.g., ESP32). The combination of the sensor and the microcontroller is termed an edge device. As a result, two attack surfaces emerge in the perception layer: the sensors themselves and the devices they use for communication. Attacks on the sensors could lead to the generation of erroneous data, which, in turn, can result in meaningless insights and even property damage. For instance, consider a system that triggers a fire sprinkler upon registering an ambient temperature greater than 200°F. An attacker could physically or remotely tamper with the temperature sensor to trigger false alarms and cause harm to the aging residents. The countermeasures outlined in Table 1.2 are not infallible. Hence, following the defense-in-depth security strategy, we require a secondary defense mechanism to detect attacks should the primary measures fail. In that regard, researchers have traditionally focused on identifying anomalies in time series data of sensors as a means to detect compro-

mises. As a result, numerous anomaly detection techniques have been proposed in the existing literature that leverage statistics and machine learning. However, they primarily focus on the time series data of each sensor in isolation. To our knowledge, no studies have explored the use of relationship trends between co-located sensors to detect anomalies. Establishing the existence of such relationships and using them in anomaly detection constitute the primary area of our research. Secondly, since the edge devices transmit their readings over a communication channel such as Bluetooth/WiFi, the chips used for communication become a crucial target for an attacker, either to compromise their availability or affect the integrity and confidentiality of the data transmitted. As seen in Table 1.1, many edge devices used in AgeTech leverage Bluetooth for communication. The recently disclosed BrakTooth vulnerabilities [11] can affect the availability of these Bluetooth-enabled edge devices by exploiting weaknesses in the implementation of Link Management Protocol (LMP) and baseband layers of the Bluetooth classic stack. The scope of most existing research in this area is limited to identifying additional vulnerabilities and exploiting them. To our knowledge, none have focused on detecting these attacks. Therefore, developing techniques and devices to address this gap constitutes the second part of our research.

1.3 Research Questions

The identified gaps in the areas discussed demand further investigation. To address them comprehensively, we've outlined the following research questions, the answers to which form the basis of this thesis.

- RQ1: Do co-located sensors in an IoT smart home exhibit linear/nonlinear correlations?
- RQ2: How can these correlations be leveraged in combination with supervised machine learning for contextual anomaly detection?
- RQ3: How can traffic at the Link Manager Protocol (LMP) and baseband layers of the BT Classic stack be effectively captured using low-cost techniques?
- RQ4: What is the efficacy of machine learning algorithms in detecting BrakTooth-based attacks on Bluetooth devices?

1.4 Thesis contributions

The solutions presented below, with their dual-faceted approach, addresses the research questions highlighted and represents our contribution to enhancing security in IoT smart homes.

- Contextual anomaly detection: Before leveraging the relationships between co-located sensors in an IoT smart home for anomaly detection, we first prove their existence by performing correlation analysis on two real-world datasets. Subsequently, we develop an experimental setup featuring commonly used sensors, including temperature, humidity, and air quality. After demonstrating linear and non-linear relationships between the time series readings of these sensors, we trigger intentional and unintentional anomalies defined in Section 2.1.1, creating a dataset containing both benign and anomalous samples. We then use statistical scores such as correlation and mutual information to model these relationships and use them in conjunction with a sliding window approach and supervised machine learning to devise a contextual anomaly detection model that outperforms traditional techniques. This contribution was published as a journal paper: *SeniorSentry: Correlation and Mutual Information-Based Contextual Anomaly Detection for Aging in Place* [12].

- Detecting BrakTooth attacks: The BrakTooth attacks target the LMP and baseband layers of the Bluetooth classic stack. Therefore, our primary objective is capturing traffic at these layers and then using machine learning for classification. We achieve this by recompiling the Bluetooth stack of a Nexus 5 device, with debugging options enabled, so that we can use a popular framework known as the InternalBlue framework [12] to patch the firmware running on its Bluetooth chipset. This procedure allows for encapsulating LMP traffic into upper-layer HCI (host controller interface) events, creating a cost-effective experimental active sniffer. Following this, we employ the BrakTooth PoC tool [13] in tandem with the ESP32WROVER kit to generate BrakTooth attacks. The sniffer captures the traffic from these attacks and other benign communications, creating a comprehensive dataset. Subsequently, we use supervised machine learning techniques on this dataset and achieve a good classification performance. This contribution was published as a conference paper: *Detecting BrakTooth Attacks* [14].

1.5 Thesis outline

The structure of this dissertation is organized as follows. In Section 2, we introduce the background and related works pertaining to the two domains of our contribution. Section 3 presents the Contextual Anomaly Detection framework, elaborating on the datasets used for correlation analysis, the experimental setup for collecting a new dataset, and our proposed detection model. In Section 4, we detail the experimental procedures and the techniques we have devised specifically for detecting BrakTooth attacks. Section 5 presents the evaluation results for our proposed techniques. Lastly, Section 6 presents the conclusions we've drawn and offers insight into potential future work in this area.

Chapter 2

Background and Related Work

2.1 Anomaly detection in IoT sensor data

2.1.1 Background

The IoT technology, especially the sensors, are being increasingly used to support aging in place [3]. With the growing reliance of older adults on such sensors, it becomes crucial to ensure that they operate optimally and that any anomalies or deviations that may suggest a problem are detected early to prevent negative consequences. Especially in smart homes designed for older adults, the detection of anomalies becomes critical, and we classify these anomalies into two types:

1. **Intentional anomalies**: They refer to deviations from normal behavior that are created deliberately, rather than occurring naturally or by chance. These anomalies are created by malicious actors with the aim of disrupting the normal functioning of the system or causing harm. Tampering with sensors, performing network attacks, and spoofing are all examples of methods that can create intentional anomalies.

2. **Unintentional anomalies**: They refer to deviations from normal behavior or expectations that are not intentionally created, but rather occur due to a variety of factors, such as sensor malfunctions, environmental interference, user error, improper installation, and power fluctuations.

Whether the anomalies are created intentionally or unintentionally, their impact on seniors is the same. For example, a malicious actor may intentionally launch a

Denial-of-Service attack on the smart home's lighting system, causing the lights to shut off and thus increasing the danger of falls or other accidents for aging residents. Similarly, if the motion detection sensor in the home malfunctions and fails to detect movement, the lights may not turn on, resulting in a similar adverse situation. Overall, the importance of anomaly detection in smart homes designed for the elderly cannot be overstated. It is essential to therefore protect both the physical and emotional safety of these individuals and ensure that they can enjoy the benefits of smart home technology seamlessly. A simple method of anomaly detection is to establish a region that represents normal behaviour and identify any observation that falls outside of this region as an anomaly. However, several factors make this approach challenging as noted by Chandola et al. [15], such as the difficulty of capturing the concept of normalcy.

Cook et al. [16] presented an alternate categorization for anomalies in time series IoT data, dividing them into three types: point anomaly, contextual anomaly, and collective anomaly. Contextual anomaly is defined as an observation that may otherwise appear to be normal, but when considered within a context, can be considered an anomaly. To construct a contextual anomaly detection model, we first need to learn or identify the contextual and behavioural attributes of the system. Contextual attributes are characteristics of the environment or context in which the sensor data is collected. For instance, in multivariate time series datasets, time can serve as a contextual attribute. Behavioural attributes are characteristics of the data that indicate its behaviour or pattern over time. One common behavioural attribute in time series data is the trend, which refers to the general direction or pattern of the data over time. Contextual anomalies can be particularly challenging to detect, as they only appear abnormal within a specific context. For example, a sudden increase in the number of steps taken by an older person in the middle of the night, as recorded by a wearable fitness tracker, could indicate an anomaly. The high number of steps in itself is not an anomaly; however, in the context of the time of the day, it is considered an anomaly. Therefore, there is a need to discover various contexts and apply them to observations to establish normal vs abnormal activity.

Hayes et al. [17] employed the idea of profiles to create contexts by grouping similar data points together using a multivariate clustering algorithm and applied it to detect contextual anomalies. Carmona et al. [18] proposed an anomaly detection framework, called neural contextual anomaly detection (NCAD) which incorporates contexts through a contextual hypersphere. In this study, we aim to use the rela-

tionships among sensors as contexts for anomaly detection. Co-located sensors often exhibit linear or nonlinear relationships [19]. Using these relationships as behavioral attributes and a sliding window of a fixed number of samples as a contextual attribute for contextual anomaly detection has not been explored to the best of our knowledge, and our study aims to fill this gap. While correlation can be used to model linear relationships, it falls short when it comes to capturing nonlinear interactions between two variables. In such scenarios, mutual information proves to be effective. Mutual information represented by eq. 2.1 quantifies the shared information between two variables, effectively measuring the reduction in uncertainty about one variable when the other is known. This makes it a powerful tool for modeling nonlinear relationships.

$$I(X;Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right) \tag{2.1}$$

Ultimately, our objective is to identify anomalies, whether intentional or unintentional, that violate the linear and nonlinear correlations among sensors using correlation and mutual information scores. Firstly, we prove the presence of linear or nonlinear relationships among co-located sensors in a smart home by performing correlation analysis on two public benchmark datasets: the open smart home dataset [20] and the smart building dataset [21] and a dataset we collected in our laboratory at the University of Victoria, called the ISOT AgeTech dataset. Secondly, we use a sliding window of a fixed number of samples (i.e., the contextual attribute) on the ISOT AgeTech dataset to determine the correlation and mutual information scores between sensor readings (i.e., behavioral attributes). Thirdly, we use these measurements as features to train selected machine learning models and evaluate their ability to detect anomalies. We apply the proposed method (i.e., the second and third steps) solely to the ISOT AgeTech dataset and not to the other datasets mentioned because it is the only dataset with benign and anomalous samples. Therefore, the public benchmark datasets are used only to substantiate the existence of correlations between sensors.

2.1.2 Related Works

Several statistical techniques for detecting outliers have been proposed in the existing literature [22, 23, 24]. Apart from the statistical techniques, there also exist density [25], distance [26], clustering [27], and machine learning-based anomaly detection techniques [28]. However, a key problem with most existing anomaly detection meth-

ods is that they tend to ignore the context in which the data is generated. This lack of consideration of context can lead to incomplete and inaccurate results [17]. Consideration of the context is particularly crucial for detecting anomalies in smart homes designed for older people.

Artola et al. [29] recognized this lack of research on contextual anomaly detection in relation to the well-being and healthcare of older adults. They proposed a system that uses a wearable device to collect data on older adults' heart rate, sleep duration, and daily step count (i.e., behavioural attributes) in relation to the time of day (i.e., contextual attribute). Then, they utilized multiple open-source anomaly detection models to identify any deviations and report them to a healthcare provider.

Shahid et al. [30] focused on building a model to learn behavioural patterns of older adults in smart homes and proposed an anomaly detection model that learns behavioural attributes, namely the amount of time spent and number of visits the resident makes to each room of the house (i.e., contextual attribute). The researchers then applied a non-parametric statistical method based on Chebyshev's inequality theorem to detect anomalies in daily user activities. They defined an outlier as an observation whose duration exceeded what was expected by two standard deviations. This method is limited by the use of Chebyshev's inequality, in which thresholds are based on loose intervals and 75% of the data fall within two standard deviations. This is less precise than a normally distributed dataset, in which 95% of the data falls within two standard deviations. Having wider intervals between thresholds can result in fewer anomalies because fewer data points fall outside the defined range.

Aran et al. [31] collected data from pressure, motion, and door sensors located in 40 different households of elderly people to model their daily behaviour and identify anomalies. They developed a probabilistic spatio-temporal model to summarize normal behaviour and used cross-entropy measures to identify and categorize significant deviations from the norm as anomalies. However, the proposed approach suffers from the unavailability of ground truth labels and the inability to generalize to multiple residents.

In the broader context of IoT smart homes, Chenglong et al. [32] introduced a semantics-aware anomaly-detection system termed the Home Automation Watcher (HAWatcher). This system models the normal behavior in smart homes by generating correlations from semantic information, such as installation locations, device types, smart apps, configurations and relations. These correlations are categorized into two types: e2e (event to event) and e2s (event to state). The HAWatcher also

contains a shadow execution engine that simulates the states of various devices based on the observed correlations. Any deviation between the simulated and real-world device states is flagged as an anomaly. Overall, it achieved an impressive precision of 97.83% and a recall of 94.12%, notably surpassing previous methods. However, since a discrete set of state transitions are considered in the context of observed correlations, only a limited type of anomalies can be detected.

Researchers so far have focused on modelling the behavior of older adults, using relevant AgeTech sensors, to perform context-based anomaly detection. However, only a few focused on modelling the relationship between such sensors to do the same. While we could not find any research that specifically uses the relationship between AgeTech sensors for contextual anomaly detection, we did find a few studies that make limited use of these relationships in a different domain.

For example, Deng et al. [19] used a graph structure learning approach to determine the relationship between the sensors associated with water treatment and water distribution. They also employed a graph attention-based forecasting method to predict the expected value of a sensor at a specific time and classified observations as anomalies if the deviation between the forecasted and actual values exceeded a threshold. However, this technique relies on prediction and does not fully consider the continuous nature of the correlation and mutual information between the sensors.

Li et al. [33] proposed a method that involves creating a temporal correlation graph by analyzing the correlation between different features in an industrial multi-sensor system and then using a specialized neural network (called a structured-sensitive graph neural network) to extract useful information from the graph, such as the relationships between points, edges, and overall structure. This information, along with preset thresholds on the fluctuations of correlation and sensor values as hyper-parameters, is then used to classify the graph and detect any anomalies. Although this technique models the linear relationship between the sensors using correlation coefficients, it cannot model nonlinear relationships between sensors.

Current methods of context-based anomaly detection usually rely on the pre-identification of both contextual and behavioural attributes, with the assumption that the context is determined by spatial or temporal characteristics. However, in reality, it can be difficult to identify the true context in a dataset, particularly when the dataset is high-dimensional and has numerous attributes that can be combined in different ways to create the context [34]. No prior methods exist, to our knowledge, that take into account both the linear and nonlinear relationships between AgeTech-

related IoT sensors to build an anomaly detection model.

2.2 Detection of Bluetooth attacks

2.2.1 Background

Bluetooth is a wireless communication standard that can be used to exchange data between stationary and fixed devices within a range of up to 100 meters [35]. Although Bluetooth occupies the ISM band of 2.4 GHz which is 83MHz wide, it does not use the Direct Sequence Spread Spectrum (DSSS) used by WiFi. Instead, it uses Frequency Hopping Spread Spectrum (FHSS) to hop between 79 different 1 MHZ-wide channels in this band. Due to its use of FHSS, interference with other devices is reduced. However, WiFi uses a single channel that is 22MHz wide, and when both WiFi and Bluetooth networks are in the same range, the 22MHz channel of WiFi occupies 22 of the 79 Bluetooth channels leading to some interference. When a Bluetooth device experiences interference, it addresses this problem by hopping to the next channel and retrying. In contrast, WiFi issues an Automatic Repeat Request and slows down the data rate in an attempt to reduce the Bit Error rate. However, the number of channels and the channel bandwidth differ based on the type of Bluetooth device used. There exist two types of Bluetooth devices: Bluetooth Classic (BT Classic) and Bluetooth Low Energy (BLE). A few salient differences between these two devices are listed in Table 5.5.

According to the Bluetooth Special Interest Group [36], it is projected that over 7.1 billion Bluetooth-enabled devices will be shipped in 2026. This figure encompasses all Bluetooth technology devices, including BT Classic and newer versions such as Bluetooth Low Energy (BLE). Although the number of devices shipped with just BT Classic is expected to decline, 100% of the future Bluetooth-enabled devices are expected to support dual-mode (BT Classic + BLE). A substantial number of these devices are used by aging adults, such as smartwatches, hearing aids, fall detection devices, blood pressure monitoring devices, weighing scales, pillboxes, diapers, and more, to assist them in everyday activities [37]. Therefore, it becomes crucial to protect these devices against attacks that intend to compromise them.

In recent years, Garbelini et al. [11] has disclosed a family of new security vulnerabilities in commercial Bluetooth stacks that can compromise the availability of BT Classic devices, called BrakTooth. The BrakTooth vulnerabilities exist especially in

Table 2.1: Salient differences between BT Classic and BLE.

Feature	**Bluetooth Classic**	**Bluetooth LE**
Application	Audio streaming and data transfer.	Audio streaming, data transfer, location services and device networks.
Frequency band	2.402 - 2.480 GHz (ISM)	2.402 - 2.480 GHz (ISM)
Channels	79	40 channels
Channel bandwidth	1 MHz	2 MHz
Spread spectrum	FHSS	FHSS
Power Consumption	1W	∼0.001 W-0.5 W
Data rate	1 Mb/s, 2 Mb/s, 3Mb/s	125 Kb/s, 500Kb/s, 2 Mb/s
Device discovery	Inquiry or paging	Advertising
Encryption algorithm	E0/SAFER+	AES-CCM
Network topology	Point-to-point	Point-to-point, Broadcast and Mesh

the link manager and baseband layers of the Bluetooth stack. Several vendors producing BT Classic system-on-chips (SoCs) were notified of the disclosures and they have already started patching their implementations of the stack, or have already patched them. Usually, vendors provide security patches to SoCs through wireless over-the-air (OTA) firmware updates, or wired updates (i.e, via USB) [38], or replace the devices with patched SoCs; the last one requires recall and is rarely done. Typically, updating the firmware of Bluetooth-only devices over the air requires two separate devices: a Device Firmware Update (DFU) target and a DFU controller. The DFU controller, which is usually a mobile device running a vendor-specific application, is responsible for transferring the firmware image to the DFU target device where the update needs to be made. Nordic Semiconductors employs this technique to update their Bluetooth-based devices over BLE [39]. Several devices used in aging in place have only Bluetooth connectivity, therefore, older adults without sufficient technical expertise find it challenging to update them. As a result, several such devices remain vulnerable to BrakTooth attacks. To be able to detect these attacks, we must be able to first sniff traffic. Several devices and tools are readily available to monitor wireless WiFi traffic. However, monitoring Bluetooth traffic reliably is limited to the realm of professional equipment like the Ellisys Vanguard or the Frontline Sodera and they can be very expensive [40]. In general, there exist two types of sniffing:

- Passive sniffing: It involves a device that sniffs over-the-air Bluetooth packets between two communicating devices in radio range.

- Active sniffing: It involves a device that connects to a remote BT device and captures the packets exchanged down to the lowest layer in the Bluetooth protocol stack.

Due to the fact that Bluetooth uses frequency-hopping spread spectrum technology with a vendor-specific hopping pattern, it is difficult to monitor its connections. Despite these challenges, a few inexpensive passive sniffers such as Ubertooth One [41] and nRF Sniffer [42] were developed, however, they can only capture packets of BLE reliably, and not BT Classic [43]. To the best of our knowledge, there is still no non-commercial solution for monitoring BT Classic traffic. One approach to capturing the packets pertaining to BrakTooth attacks would be to set up an Android device as the active sniffing target and enable a debug functionality in its Bluetooth stack known as the HCI Snoop Log feature [44]. This feature captures all BT traffic above the HCI layer of the Bluetooth stack. However, as shown in Figure 2.1, since the BrakTooth attacks are aimed to exploit the vulnerabilities in the link manager protocol (LMP) and baseband layers of the Bluetooth stack, we need a method to monitor the lower-level traffic. Thus, we adopted a well-known framework called InternalBlue [45], which is capable of monitoring communications below the HCI layer. The researchers who developed this framework reverse-engineered widely used Broadcom BCM4339 Bluetooth Controller firmware and developed patches that give access to the LMP layer. Note that the Nexus 5 is one of the popular mobile devices that contain the BCM4339 controller [46]. In this study, we utilized the InternalBlue framework to enable LMP monitoring on a Nexus 5 device and use it as an active sniffer to capture LMP packets from both benign and malicious devices. It should be noted that this approach to capturing LMP packets was not explored in prior works. As a consequence, detecting BrakTooth-based attacks using machine learning and inexpensive hardware was not feasible until now. Therefore, our study presents only our findings without a comparative analysis.

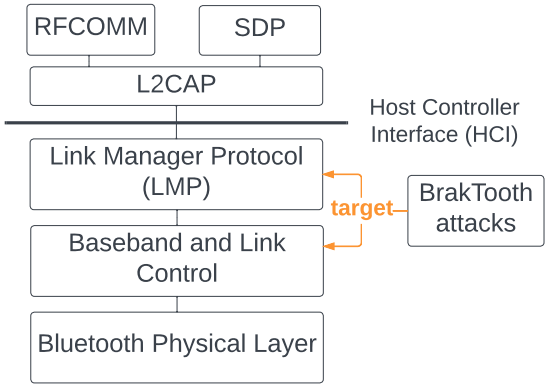

Figure 2.1: BrakTooth attacks targeting the LMP and baseband layers of the BT Classic stack.

The following section discusses some works that use both commercial and non-commercial devices to detect attacks utilizing various methods such as thresholds in Bluetooth specifications, signatures, statistical distributions of data, and anomaly detection.

2.2.2 Related Works

Wu et al. [47] used the Ubertooth One device to capture over-the-air packets in the BLE advertising channel and extracted features such as the advertising pattern, state transitions of the BLE device, advertising interval, the RF (Radio frequency) signal frequency offset, and the RF signal strength. They detect spoofing attacks by creating a statistical distribution of CFO (Carrier frequency offset) and RSSI (Received signal strength indication) values and use them to identify any value that falls out of a predetermined threshold. They first look for a connection request packet to identify the detection of a new device and then use the above statistical distributions to detect a spoofing condition. One drawback of using the Ubertooth One device for BT classic monitoring is that it is unable to capture all packets reliably [43, 48].

Connor et al. [49] presented a Bluetooth-based network intrusion detection system that uses the Merlin LeCroy protocol analyzer to non-intrusively capture the Bluetooth traffic and detect malicious behavior using pattern matching. It decodes Bluetooth packets by synchronizing with the master device on a piconet and following the hopping sequence of that piconet. However, since the detection is signature-based, novel intrusions cannot be detected. The authors have used a commercial protocol

analyzer and hence were able to observe and analyze up to the LMP layer of the Bluetooth stack.

Huang et al. [50] used loopholes in the Bluetooth specification pertaining to low-power mode transitions, to perform Bluetooth DOS attacks. Particularly, they exploited loopholes in the sniff and park mode conversions to trigger DOS conditions, using slave devices in a piconet. They also suggested that DOS conditions can be triggered intentionally or unintentionally when multiple piconets pertaining to different devices come closer to each other. Because devices operating in different piconets use their own frequencies for communication and are unaware of each other's hopping sequences, they may end up causing interference which can lead to a DOS condition. To detect DOS conditions caused by interference, they compare quality characteristic data (i.e, bit error rate and invalid data rate) of all the channels with the normal thresholds in the Bluetooth specification. To detect DOS attacks by slave devices, they compared the proportion of data transmission time to the slave timeslots, under normal conditions and during the activity under observation.

Satam et al. [51] proposed an intrusion detection system that runs on a Linux server machine and captures the Bluetooth data frames like SCO (Synchronous Connection Oriented) data frame, the HCI protocol frame, and the HCI (Host controller interface) data frame. Although the paper does not mention the hardware used to sniff the Bluetooth traffic, it appears that the authors used the machine's stock Bluetooth adapter because captured frames represent packets above the HCI layer. Once captured, the frames are converted into flows of size T seconds, which are then converted to n-grams. These n-grams are used to train various machine learning algorithms to establish normal behavior and thus classify a new observation flow as normal or abnormal. They performed Bluesnarfing and power-draining attacks and achieved precision and recall of 99.6%.

Although the aforementioned works employ various novel techniques to detect different types of Bluetooth attacks, to the best of our knowledge, there is currently no existing research that focuses on the detection of BrakTooth-based Bluetooth attacks using affordable hardware and machine learning techniques. Our work aims to fill this gap.

2.3 Summary

In the area of anomaly detection in IoT smart homes, as mentioned in Section 2.1.2, although some prior work dealt with monitoring the behaviorial characteristics of older adults and identifying various context through the use of sensors, no work considered the relationship between sensors themselves to identify intentional and unintentional anomalies in sensor readings. In this work, we first confirm the existence of relationships between co-located sensors through experiments and data analysis. We then model these relationships using statistical measures such as correlation and mutual information scores to establish a normal baseline. Finally, we use a sliding window technique in combination with supervised machine learning to detect anomalies.

In the area of detection of Bluetooth attacks, as mentioned in Section 2.2.2, most of the prior works focused on spoofing conditions through statistical distributions, detecting malicious behavior through rule based techniques and flagging intrusions through supervised machine learning. Furthermore, due to lack of inexpensive devices that capture traffic below the LMP layer of the Bluetooth stack, most works captured packets above LMP layer or used basic bluetooth features such as carrier offset, and the RF signal strength. In this work, we use the InternalBlue [45] framework with the Nexus 5 device to develop an inexpensive active sniffer that can reliably capture traffic below the LMP layer. We then use this in combination with supervised machine learning to detect Braktooth attacks with high degree of accuracy.

Chapter 3

Contextual Anomaly Detection Framework

3.1 Datasets

Two major perspectives can be emphasized when collecting AgeTech datasets: infrastructure and activity. The infrastructure perspective defines the environment in which a senior person lives, while activity captures their daily routines. Although these perspectives are interdependent, one can be emphasized over the other in data collection. Similarly, our proposed detection model focuses primarily on the infrastructure aspects. A combination of different sensor types can be deployed in a typical AgeTech environment. Table 1.1 provides several examples of common sensors that can be found in AgeTech smart homes. After an extensive search, we could not find publicly available IoT-sensor datasets related to aging in place. As an alternative, we utilized the open smart home, smart building and ISOT AgeTech datasets to substantiate the existence of correlations between co-located sensors in a smart home. Subsequently, we used the ISOT AgeTech dataset solely to validate our model, as only it has both benign and anomalous samples. In the following sections, you will find an overview of these three datasets.

3.1.1 Open Smart Home Dataset

The open smart-home dataset was collected at Fraunhofer Institute for Building Physics, Nürnberg, Germany by Schneider et al. [20]. It contains time series measurements of temperature, brightness and humidity sensors placed in the bathroom,

kitchen, rooms 1, 2 and 3 and the toilet of a smart home located in this building. The placement of the various sensors is shown in Table 3.1 and the corresponding counts of the data samples pertaining to each sensor are shown in Table 3.2.

Table 3.1: Placement of the sensors in different rooms for the open smart-home dataset.

Location	**Brightness**	**Humidity**	**Temperature**	**Thermostat**
Bathroom	1	1	1	1
Kitchen	1	1	1	1
Room 1	1	1	1	1
Room 2	1	1	1	1
Room 3	1	1	1	2
Toilet	1	1	1	1

Table 3.2: Number of data samples per sensor type in the open smart-home dataset.

Type	**Counts of Samples**
Brightness	64,629
Humidity	10,076
Temperature	81,029
Thermostat Temp.	74,046

3.1.2 Smart Building Dataset

The smart-building dataset was collected in Sutardja Dai Hall (SDH) at UC Berkeley by Hong et al. [21]. It contains time series measurements from 255 sensors that were placed across 51 rooms. Each room had a CO_2-concentration sensor, an air-humidity-measurement sensor, an air-temperature-measurement sensor, a luminance sensor and a passive-infrared-ray (PIR) sensor. The distribution of data samples per sensor type is listed in Table 3.3.

Table 3.3: Distribution of data samples per sensor type in the smart-building dataset.

Sensor Type	Counts of Samples
Luminance	6,571,412
Temperature	6,571,454
CO_2	6,573,957
PIR	3,593,902
Humidity	6,571,414

The timestamps of the collected data samples were in Unix Epoch Time. All sensors were sampled once every 5s, while the PIR motion sensor was sampled once every 10s. The PIR sensor helps determine the presence of a person inside a room by measuring the radiation emitted from the subjects in its proximity.

3.1.3 ISOT AgeTech Dataset

Temperature, humidity and air-quality sensors are commonly used in homes designed for older adults to monitor the indoor environment and ensure that it remains safe and comfortable for residents. In addition, they are readily available and highly dependable, which is why they were utilized in collecting the ISOT AgeTech dataset. The dataset contains sensor data collected from two DHT22 and two MQ135 sensors located in the ISOT laboratory at the University of Victoria. Each DHT22 sensor contains one built-in temperature sensor and one built-in humidity sensor. The descriptions of the DHT22 and MQ135 sensors are given in Table 3.4.

Table 3.4: Descriptions of DHT22 and MQ135 sensors

Sensor	Description
DHT22	The DHT22 is a low-cost digital temperature and humidity sensor.
MQ135	MQ135 is an air-quality sensor that is extremely sensitive to benzene, sulfide, smoke and other harmful gases.

Data Collection Network Architecture

The floor plan of the laboratory and placement of the sensors is shown in Figure 3.1. The collection network architecture consists of three components: IoT sensors, a fog

node and a cloud server. The fog node is responsible for aggregating information from different IoT sensors over a certain time and sending it to the cloud server. Sensor information is posted by the corresponding micro-controller unit (ESP32) to the fog node using HTTP REST protocol and the cloud server provides data storage and eventually data-processing capability (e.g., using machine-learning models). In total, we collected data from the following six sensors:

1. Two temperature sensors (DHT22).
2. Two humidity sensors (DHT22).
3. Two air-quality-measurement sensors (MQ135).

Each sensor sends its data every 30s to the Web API hosted on the fog node (a machine in the ISOT laboratory). A new file is created for each such sensor and uploaded to our GitHub repository once every day [52], as shown in Figure 3.2. The datasets starting with keywords 3R32 and 3U38 contain the temperature and humidity information collected from the two DHT22 sensors, whereas the datasets starting with keywords 3U46 and 3U48 contain the air-quality information sourced from the two MQ135 sensors. The top five rows of the dataset pertaining to the temperature and humidity sensor (3R32) are listed in Table 3.5.

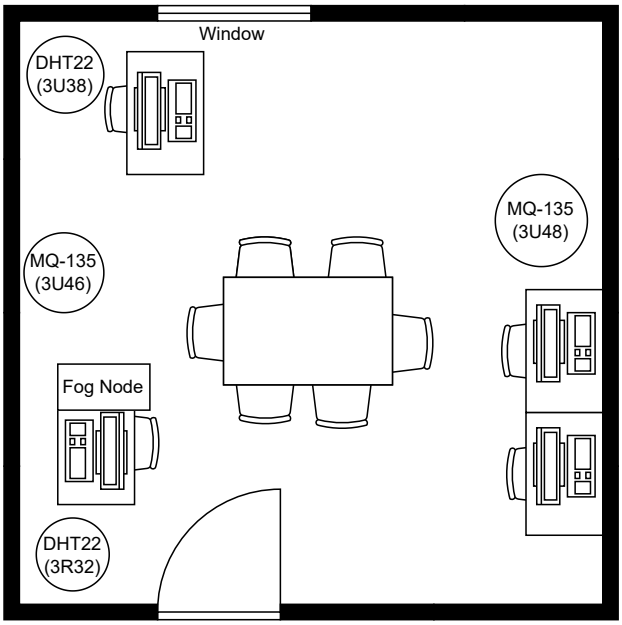

Figure 3.1: Floor plan and placement of the sensors in the ISOT laboratory.

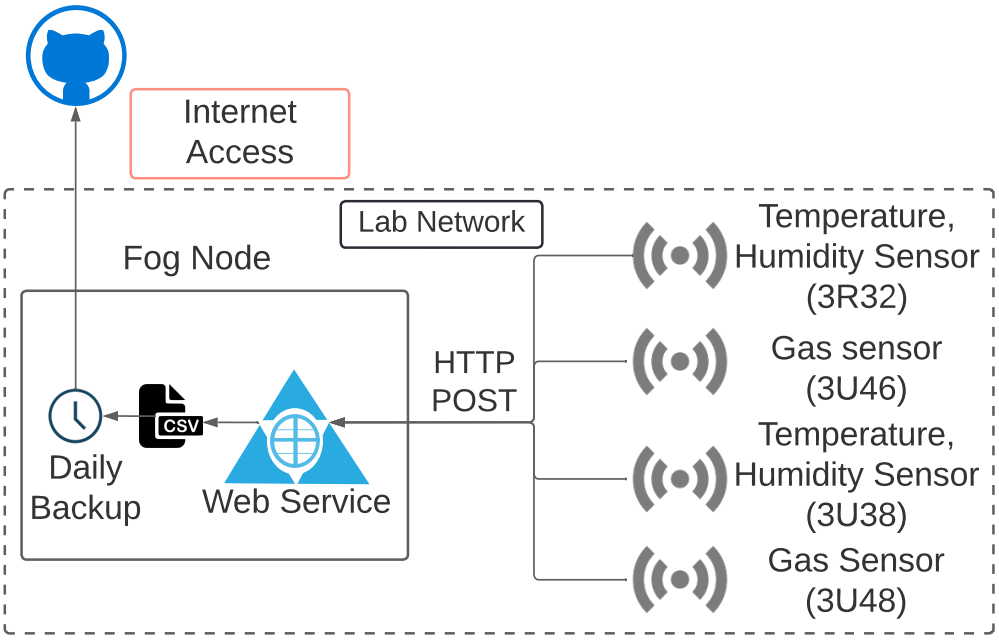

Figure 3.2: ISOT laboratory setup.

Table 3.5: Examples of temperature and humidity sensor (3R32) data collected in the ISOT AgeTech dataset.

Timestamp	Temperature	Humidity	Type
1662689838001	44.09999851	20.29999924	normal
1662689868083	44.09999843	20.29999922	normal
1662689898161	44.09999847	20.29999926	normal
1662689928240	44.09999847	20.29999923	normal
1662689958321	44.20000076	20.20000036	normal

3.1.4 Exploring Normal and Anomalous Data

Both the open smart-home and smart-building datasets consist solely of normal data. The sensors and procedures used to collect this data are outlined in Sections 3.1.1 and 3.1.2. In contrast, the ISOT AgeTech dataset includes both normal and anomalous samples, which are detailed in the following sections.

3.1.5 Normal Data Of ISOT AgeTech Dataset

The normal samples in the ISOT AgeTech dataset represent observations captured during regular human activity in the experiment room of the ISOT laboratory. During the capture period, five individuals worked in the room from 9 AM to 8 PM and the room remained empty overnight. All sensors were placed next to the individual carrels and there was constant human traffic in and out, which was captured by the sensors. It is well known that human presence can affect the ambient temperature, humidity [53] and CO_2 [54] in enclosed spaces. The distribution of the sensor data under normal conditions is presented in Figure 3.3. A subsequent Shapiro-Wilk test conducted on this data confirmed its non-normal distribution.

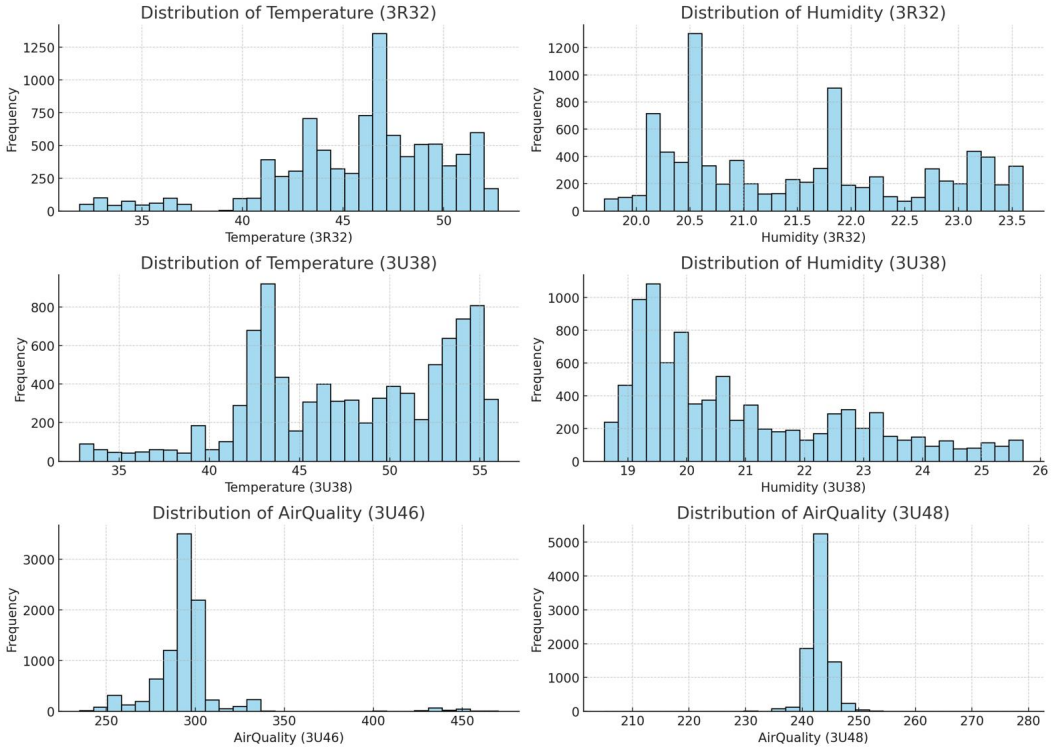

Figure 3.3: Distribution of sensor data collected between September 4, 2022 to September 9, 2022.

3.1.6 Anomalous Data Of ISOT AgeTech Dataset

As explained in Section 2.1.1, in the context of aging-in-place smart homes, anomalies can be categorized into two types: intentional and unintentional. Their roles in the

ISOT AgeTech dataset are presented below:

Intentional Anomalies: As a conduit to produce intentional anomalies, we conducted two types of network-based attacks on the web service hosted on the fog node.

1. Distributed denial-of-service (DDOS) attacks: These attacks were conducted to disrupt the collection of sensor readings and effectively disable the fog node. To achieve this, we used an open-source tool called the PYbot Botnet [55].

2. Replay attacks: These attacks were executed by altering the payloads and replaying the previously captured HTTP requests sent by various sensors to the fog node.

A malicious actor can choose to use randomly generated sensor data during these attacks, but this is a simplistic and unsophisticated method that can be quickly identified via our model, as randomly generated sensor data readily violates sensor relationships. A more natural approach is to use data that resembles sensor data. Therefore, we chose to source these sensor values from the attack data of another IoT-sensor dataset called the TON_IoT weather dataset [56, 57, 58, 59, 60, 61, 62] which was generated from the same type of sensors as the ISOT AgeTech dataset. Therefore, it is important to note that our model does not necessarily detect attacks but the anomalous conditions that these types of network attacks may create, which violate the correlation between sensors.

Unintentional Anomalies: These anomalies are typically caused by sensor malfunctions, incorrect installation by the user, improper usage or environmental interference. To simulate such anomalies, we altered the sensitivity of the MQ135 sensors by intermittently adjusting the potentiometers to their maximum or minimum values and periodically disconnecting the wiring for the DHT22 sensors. This resulted in abnormal sensor values that can serve as representative examples of unintentional anomalies. Gaddam et al. [63] referred to such anomalies as intermittent sensor errors and binary failures. Because the data from various sensors reside in different files, we first merged all of the files collected between 4 September 2022 and 9 September 2022. All the sensor readings within the range of 30 s were merged into a single timestamp. Table 3.6 shows the top five rows of the merged dataset. At a particular time, if one or more sensors are anomalous, the entire row would be labelled an anomaly. Table 3.7 shows the breakdown of normal and anomalous samples in the merged dataset. The full dataset can be accessed at [64].

Table 3.6: First five rows of the merged ISOT AgeTech dataset.

Time	Temp. (3R32)	Hum. (3R32)	Temp. (3U38)	Hum. (3U38)	AirQ. (3U46)	AirQ. (3U48)	Type
1662274800	49.09	21.0	49.70	20.70	290	238	benign
1662274830	49.09	21.0	49.70	20.79	291	243	benign
1662274860	49.20	20.9	49.70	20.70	290	242	benign
1662274890	49.20	20.9	49.78	20.70	289	244	benign
1662274920	49.20	20.9	49.77	20.70	273	245	benign

Table 3.7: Distribution of benign and anomalous samples in the merged ISOT AgeTech dataset.

Type	Number of Samples
benign	8550
intentional anomalies	2190
unintentional anomalies	418

Traditionally, anomalies are characterized as observations that significantly deviate from normal data. Although *distinct* is a subjective term, in the context of this work, we define anomalies as observations that disrupt the conventional correlations observed between sensors. As seen in this section, the anomalies we induced in the laboratory setting are more generalized, and may not encompass the full spectrum of potential real-world anomalies. These could include sensor drift, characterised by a gradual deviation of a sensor's readings over time, as well as network issues, which might manifest as delays or loss of data due to network problems. Consequently, this could leave the model inadequately prepared to distinguish between certain types of anomalies. Therefore, it is advisable to utilise this work in tandem with other detection models to augment the overall performance of anomaly detection.

3.2 Proposed Detection Model

The proposed detection model utilizes correlation and mutual-information scores between the readings of various co-located sensors as features (i.e., behavioural attributes). Additionally, a sliding window of a fixed number of samples (i.e., the

contextual attribute) is employed to extract these features. For classification, we trained three different machine-learning models and evaluated their ability to detect contextual anomalies in sensor data. As illustrated in Figure 3.4, the proposed model processes the data through a series of components outlined below, before employing supervised machine learning for classification.

- Data preprocessor: This component merges the values from various sensors against a single timestamp using a tolerance of T seconds. Additionally, it removes rows containing any missing values.
- Sliding window: This provides a context within which statistical scores are calculated.
- Statistical score computer: This component calculates the correlation and mutual-information scores between sensors over S rows within each window, creating a single new row in the transformed dataset. Consequently, each row in the transformed dataset represents the statistical relationship between sensor values across S rows of the original dataset.
- Supervised machine learning: Machine-learning classification models are subsequently trained with the transformed dataset to develop an anomaly-detection model.

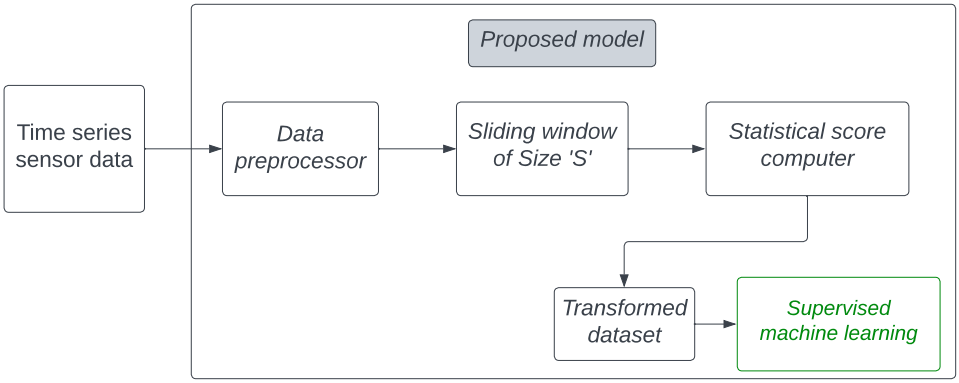

Figure 3.4: Components of the proposed method.

3.2.1 Exploratory Data Analysis

Before attempting to use statistical scores such as correlation and mutual information to model the relations, we must first address the RQ1 (Do co-located sensors in an IoT

smart home have linear or nonlinear correlations?). The open smart-home dataset comprises readings from 25 physical sensors positioned throughout various rooms of the smart home, as shown in Table 3.1. Additionally, we noted a variation in the collection frequencies of these sensors. To address this discrepancy, we opted for a 5-second tolerance while merging the sensor values into a unified dataset, proceeding to eliminate any rows with *NAs*.

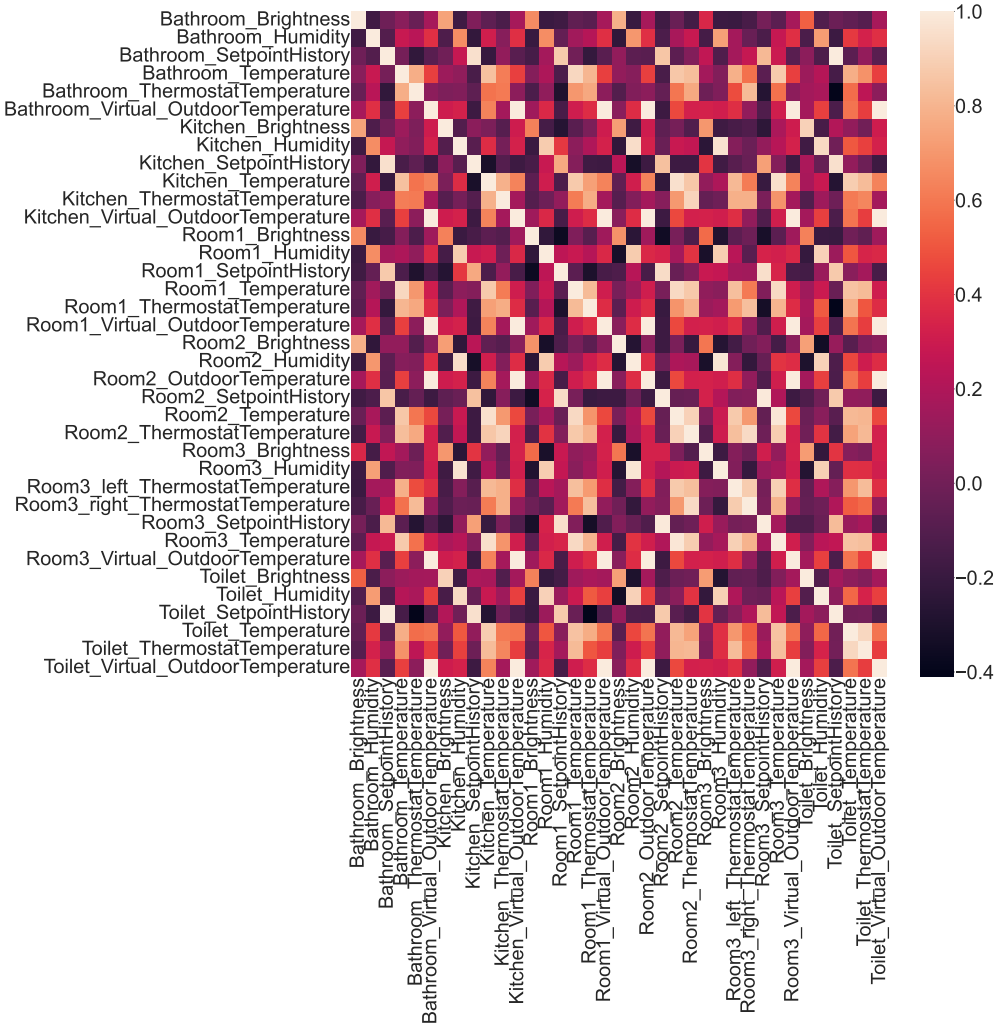

Figure 3.5: Correlation heatmap representing the correlations among the sensors in the open smart-home dataset.

The dataset features data from six virtual sensors that capture the air temperature outside each room as obtained from a virtual weather service. To test whether the values of any of these co-located sensors were correlated, we used Pearson's corre-

lation coefficient to construct the correlation matrix shown in Figure 3.5. The lighter blocks in the matrix indicate a positive correlation, while the darker blocks indicate a negative correlation. Valuable insights can be drawn from the heatmap by examining the strength of these correlations.

Table 3.8: Interpretation of Pearson's correlation coefficient.

Range	**Level**	**Range**	**Level**
0.8 to 1.0	Very Strong Positive	−1.0 to −0.8	Very Strong Negative
0.6 to 0.79	Strong Positive	−0.79 to −0.60	Strong Negative
0.4 to 0.59	Moderate Positive	−1.0 to −0.8	Moderate Negative
0.2 to 0.39	Weak Positive	−0.39 to −0.20	Weak Negative
0.00 to 0.19	Very Weak Positive	−0.19 to −0.01	Very Weak Negative

As mentioned by [65] and shown in Table 3.8, two variables are very strongly positively correlated if the coefficient value is between 0.8 and 1. Similarly, two variables are very strongly negatively correlated if the coefficient value is between -1 and -0.8. We used this reference range to extract the sensor pairs that were very strongly positively or negatively correlated and obtained the following results:

1. A total of 85 sensor pairs were found to be very strongly positively correlated; 30 of them are shown in Figure 3.6.

2. No strong negatively correlated sensors were found.

The scatter plot shown in Figure 3.7 reveals the linear relationship between the temperature sensors in the kitchen and room 2. Similarly, a linear relationship between the brightness sensors in the kitchen and toilet can be seen in Figure 3.8. Thus, we established the existence of a spatial and temporal correlation between some co-located sensors in a smart home. It is also interesting to note that these relationships may also indicate a physical connection between different rooms in a smart home. For example, the correlation between brightness or temperature sensors in the kitchen and room 2 may also suggest that the kitchen and room 2 are physically connected. However, this idea is yet to be investigated.

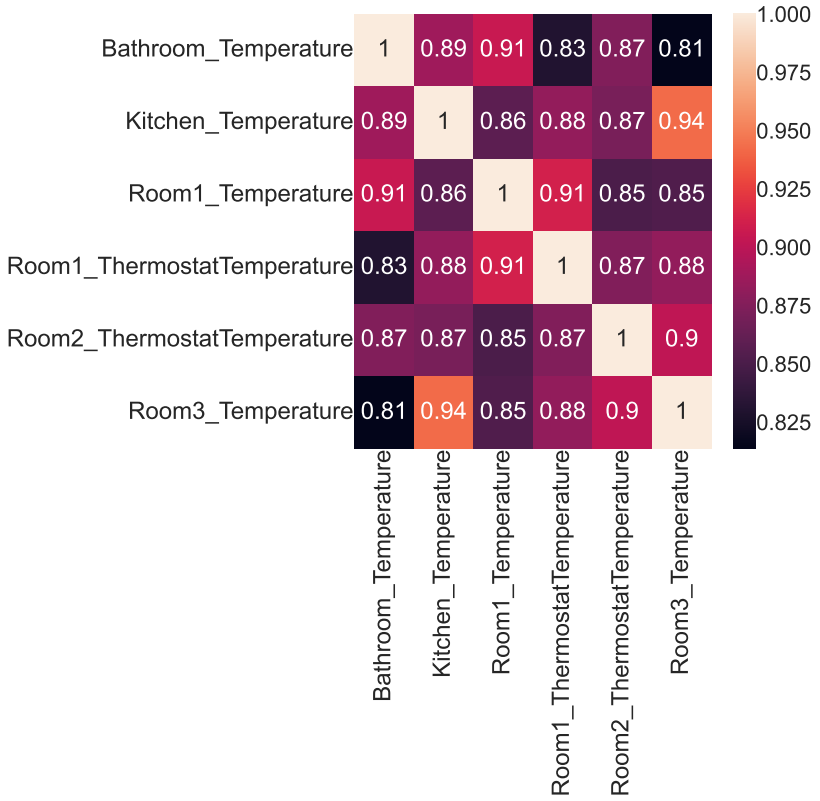

Figure 3.6: Correlation heatmap representing the correlation among 30 very strongly positively correlated sensor pairs in the open smart-home dataset.

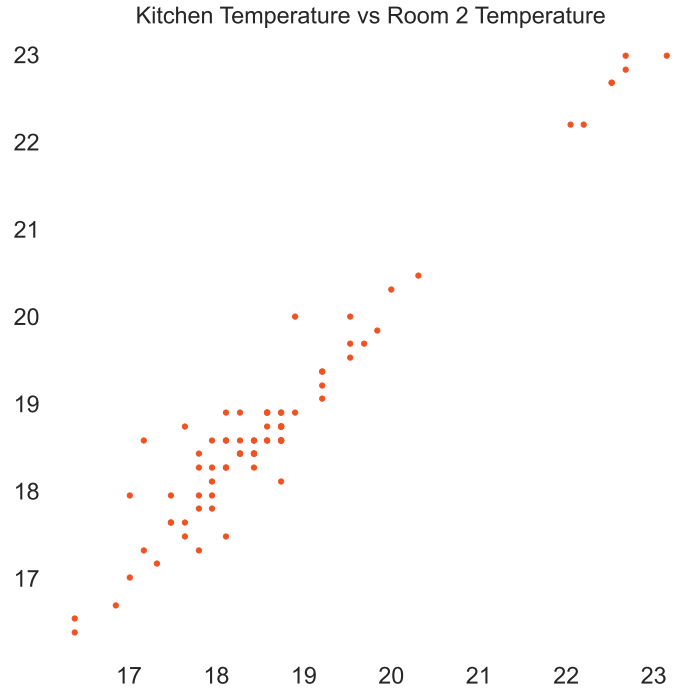

Figure 3.7: Linear relationship among temperature sensors in the kitchen and room 2 of the open smart-home dataset.

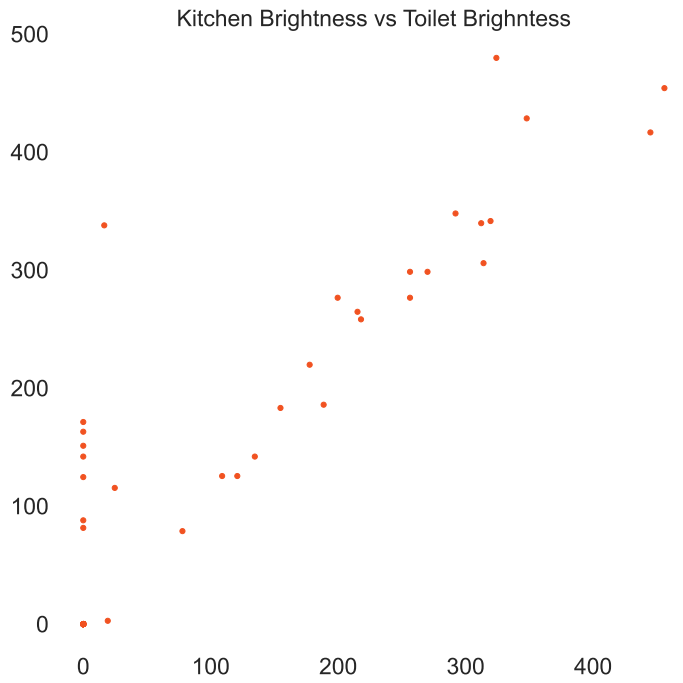

Figure 3.8: Linear relationship of brightness sensors in the kitchen and toilet of the open smart-home dataset.

Similar to the open smart-home dataset, we used Pearson's correlation coefficient to identify the strength of linear dependencies among the sensors of the ISOT AgeTech dataset. The resulting correlation heatmap, corresponding to September 9, 2022, is presented in Figure 3.9. We then extracted the very strongly positively or negatively correlated sensor pairs using Pearson's coefficient interpretation, presented in Table 3.8. We found that the temperature and humidity sensors were correlated negatively as shown in Figure 3.10 and humidity and air-quality sensors were found to maintain a nonlinear relationship as shown in Figure 3.11. While mutual information can be used to capture this nonlinearity, it falls short of offering specific insights into the nature of the relationship, be it linear, quadratic, exponential, or another type. Consequently, while we can ascertain the presence of intricate non-linear relationships between co-located sensors, pinpointing the exact nature of these relationships remains a challenge.

The same analysis was performed on the smart-building dataset and we found a total of 67 sensors that were positively correlated very strongly, out of which the scatter plot between CO_2 sensors in rooms 644 and 726 and temperature sensors in rooms 717 and 721 are shown in Figures 3.12 and Figure 3.13, respectively.

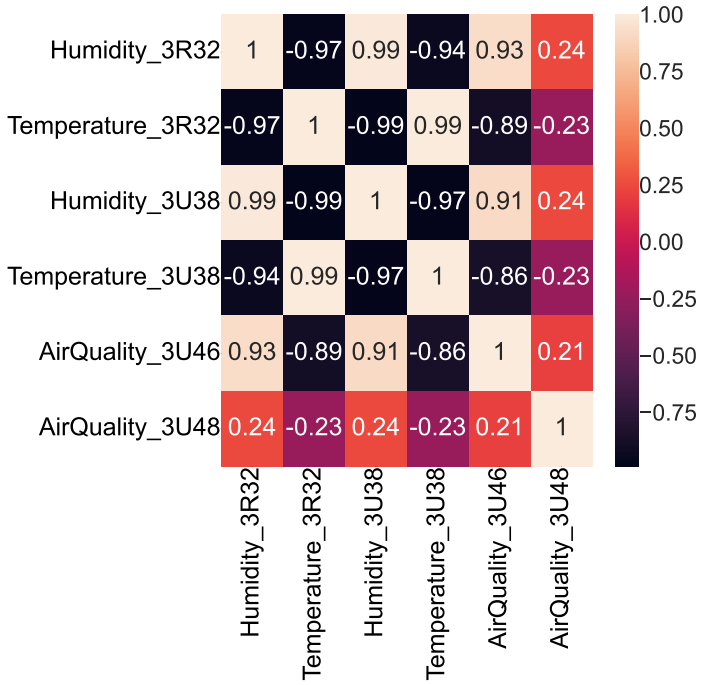

Figure 3.9: Correlation heatmap among six sensors in the ISOT dataset.

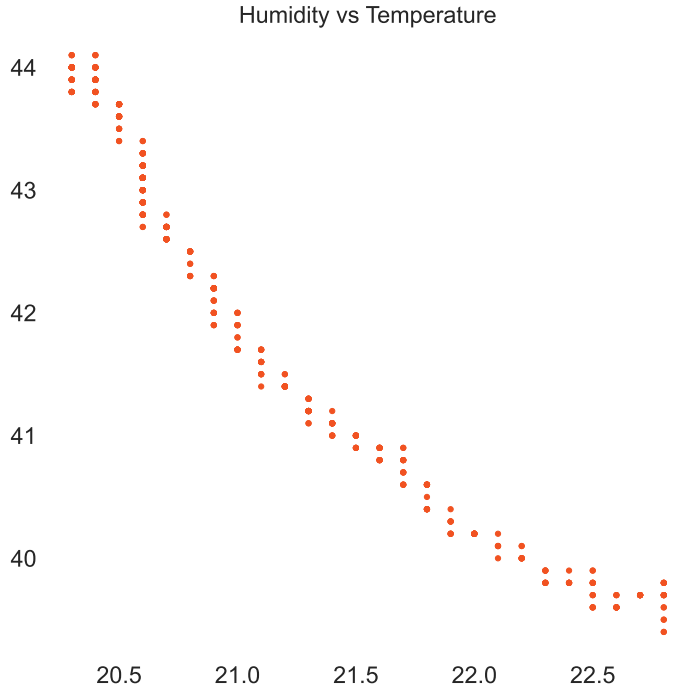

Figure 3.10: Negative correlation of temperature and humidity sensors in the ISOT AgeTech dataset.

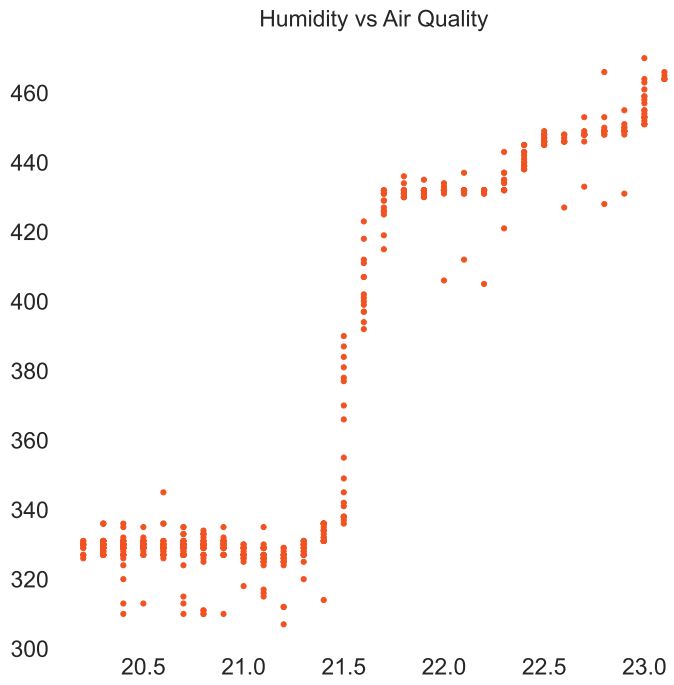

Figure 3.11: Nonlinear relationship between humidity and air-quality sensors in the ISOT AgeTech dataset.

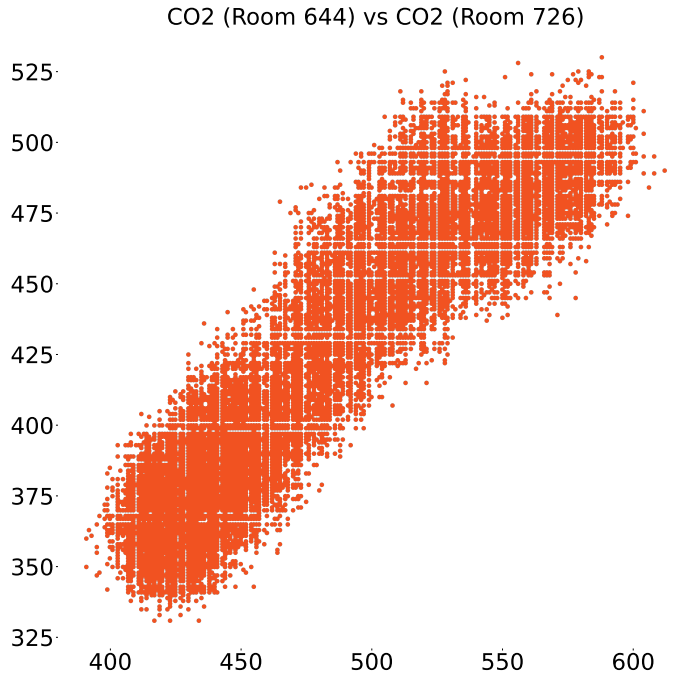

Figure 3.12: Positive correlation of temperature sensors in rooms 717 and 721 of the smart-building dataset.

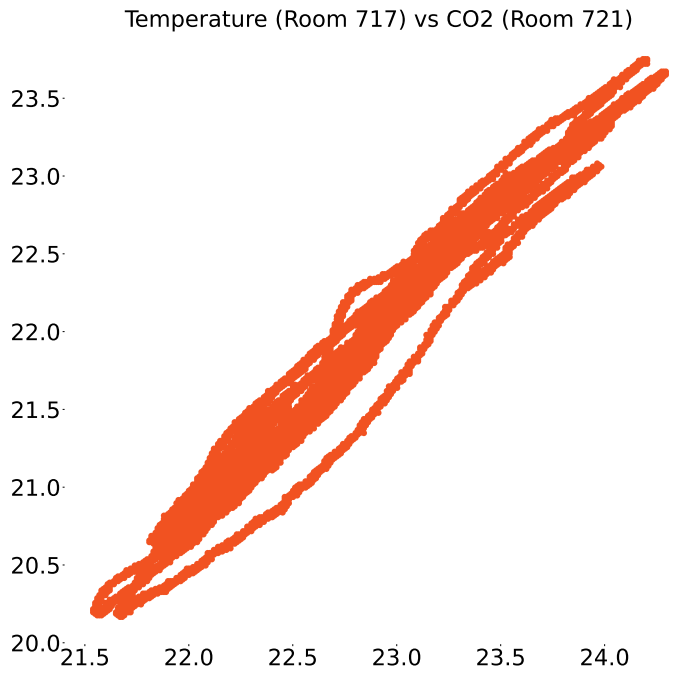

Figure 3.13: Positive correlation between CO_2 sensors in rooms 644 and 726 of the smart-building dataset.

3.2.2 Feature Model

One of the key concepts behind the proposed technique is to create new features that capture the relationships between sensors in an IoT-sensor dataset, from the existing features that represent sensor values. To explain this in more detail, let us consider the example of the ISOT AgeTech dataset and its features. From the raw sensor data of this dataset, using the sliding window approach shown in Figure 3.14, we extracted a total of 30 features by calculating the correlation and mutual-information scores between the sensor pairs. In this manner, one sample with these new features was created for each of the old (or raw) samples that fall within the range of the sliding window. If the range contained any samples with a type that was not benign, the newly created sample was labelled as anomalous.

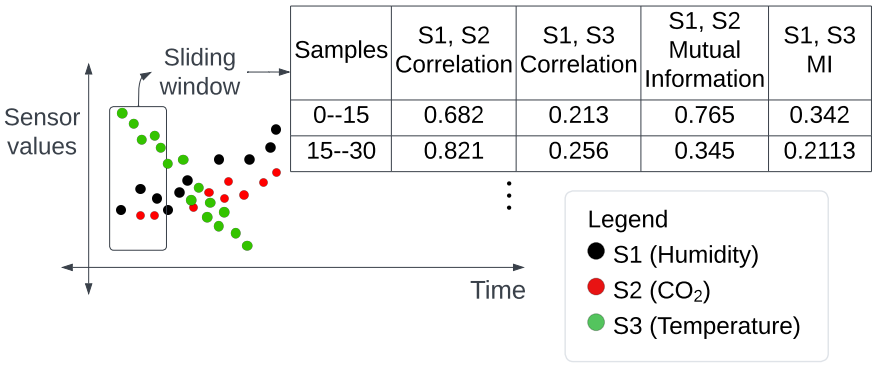

Figure 3.14: Sliding window to extract correlation and mutual information values.

3.2.3 Classification Models

Using our proposed feature model, we explored three different machine-learning classifiers: random forest, naive bayes and k-nearest neighbours (k-NN).

Random Forest: Random forest is a machine-learning algorithm used for classification and regression tasks. It is an ensemble method, meaning that it combines the predictions of multiple models to make a final prediction. In a random forest, a large number of decision trees are trained and their predictions are combined through majority voting. Each decision tree is trained on a randomly selected subset of the data and the final prediction is made by taking the majority vote (in classification tasks) or the average (in regression tasks) of the predictions made by the individual decision trees.

K-nearest neighbours algorithm: The k-NN algorithm is a simple, non-parametric method used for classification and regression. It is an instance-based learning algorithm, meaning that it does not learn a model from the training data but instead stores training data and makes predictions based on similarities in new data points to the stored training data. In the k-NN algorithm, a new data point is classified or regressed based on the majority class or average value of its k nearest neighbors, where k is a positive integer that is specified by the user. The nearest neighbours are determined based on a distance measure, such as Euclidean distance or Manhattan distance.

Naive Bayes classifier: Naive Bayes classifier is a probabilistic algorithm that uses Bayes' theorem to classify data points. Bayes' theorem describes the probability of an event occurring based on prior knowledge of conditions that might be related to

the event. In the case of a Naive Bayes classifier, the event we are interested in is a class label and the prior knowledge is represented by the probabilities of certain features (also known as predictors or attributes) being associated with each class.

3.3 Summary

In this section, we undertook exploratory data analysis on three datasets: two public benchmarks – the open smart-home dataset [20] and the smart-building dataset [21] – and one dataset we gathered at the University of Victoria, named the ISOT AgeTech dataset. Our analysis confirmed that co-located sensors typically exhibit correlations, effectively addressing RQ1. Subsequently, we employed the sliding window with a fixed number of samples (i.e., the contextual attribute) on the ISOT AgeTech dataset to determine the correlation and mutual-information scores between sensor readings (i.e., behavioral attributes). In Section 5.2.1, we use these measurements as features to train selected machine-learning models and evaluate their ability to detect anomalies. Our proposed method is applied exclusively to the ISOT AgeTech dataset. The rationale behind this is its unique presence of both benign and anomalous samples, a feature absent in the other datasets. As such, the public benchmark datasets serve primarily to validate the correlation phenomena among sensors.

Chapter 4

BrakTooth Attack Detection

4.1 Experimental Setup

The LMP protocol is used by the link managers in different Bluetooth devices to establish links and exchange information about supported features, power-saving options, and encryption techniques. Most existing works that use inexpensive, off-the-shelf Bluetooth adapters could only observe packets above the Host Controller Interface (HCI) layer, as software running on the lower layers is not accessible to the public [43]. To address this, we developed the experimental procedure seen in Figures 4.1 and 4.2 to capture the LMP traffic of normal data pertaining to benign Bluetooth communication and attack data pertaining to the exploitation of BrakTooth-based vulnerabilities. For this purpose, we used two devices:

1. Attack device: This device was used to craft and send BrakTooth attack packets to the victim device.

2. Victim device: This device received packets from the attacking device as well as normal Bluetooth packets from communication with other benign devices.

More details about these two types of devices are provided in the following subsections.

4.1.1 Attack Device

To exploit BrakTooth-based vulnerabilities, a proof-of-concept (PoC) tool [13] was released. As shown in Figure 4.1, this tool is installed on a Ubuntu 18.04 host machine

and is used to flash custom firmware on an ESP32-WROVER-KIT to execute various exploits.

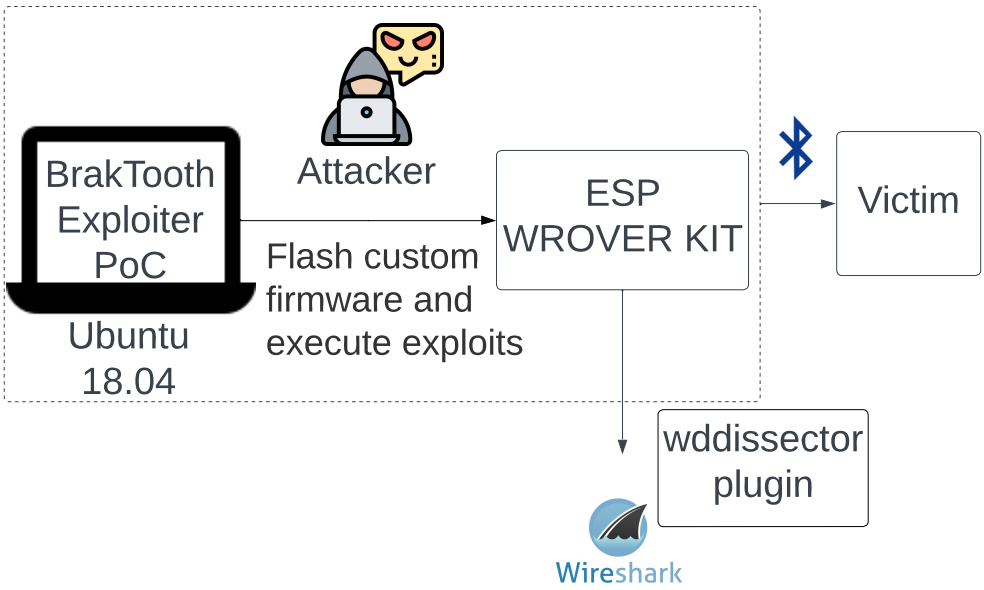

Figure 4.1: Setup of the attack device.

4.1.2 Victim Device

A rooted Nexus 5 mobile device containing the Broadcom BCM4339 Bluetooth controller is used as a victim while a Ubuntu 22.04 host machine is used to run the InternalBlue framework. The host machine connects to the victim device via Android Debug Bridge (ADB). When the victim's Bluetooth stack is compiled with debugging enabled, two new ports are opened by the Android device as follows:

1. TCP 8872: This allows for an external agent to read the HCI packets exchanged between the BCM4339 controller and the Bluetooth stack.

2. TCP 8873: This can be used to send HCI commands to the BCM4339 controller through the Bluetooth stack.

As mentioned in Section 2.2.1, the InternalBlue framework helps in monitoring the LMP packets received by the victim device and it does so by modifying the BCM4339 firmware such that it relays the LMP packets it received to the host machine over TCP 8872. The framework also uses the Bluetooth H4 Broadcom Wireshark plugin to

parse LMP and baseband packets from the incoming traffic and write to Wireshark. A setup of the victim device can be seen in Figure 4.2.

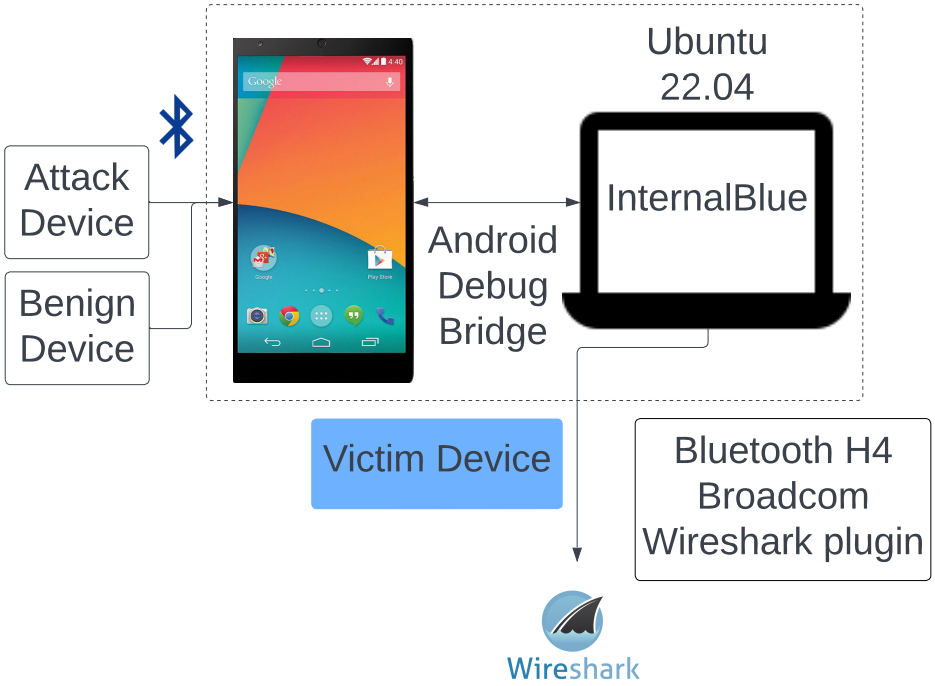

Figure 4.2: Setup of the victim device.

4.1.3 Dataset Collection

Both normal and attack data were extracted from the Wireshark capture files, generated by the InternalBlue framework running over the Nexus 5 device. To generate normal data, three mobile devices and a Windows laptop were paired with the Nexus 5 device, and they were allowed to exchange several sample documents, video, and audio files over the BT Classic protocol. To generate attack data, we used the PoC tool running on an Ubuntu 18.04 host machine to execute various exploits listed by [11], using the ESP-32-WROVER-KIT. The Nexus 5 device was kept as the target. Following is a sample command to execute exploits using the PoC tool.

```
sudo bin/bt_exploiter --host-port=/dev/ttyUSB1 --target=<target bdaddress>
--exploit =<exploit name>
```

The captured Bluetooth traffic has the following features:

1. Protocol: This field refers to the protocol used in the Bluetooth packet such as L2CAP, OBEX, Service Discovery Protocol (SDP), RFCOMM, and others.

2. Info: This field provides additional information about each packet depending on the type of protocol used.

3. Length: This field indicates the length of the packet in bytes.

4. Delta: This field indicates the time difference between the current packet and the previous packet in the pcap file.

5. Type: This field is a target variable representing normal vs attack conditions and is labeled manually.

The `LabelEncoder` was imported from the *scikit-learn* library to encode the *Protocol* and *Info* features and a standard scaler was used to standardize all the features. The distribution of the attack and normal data is shown in Table 4.1. As the number of features is limited, in this paper, we focus on classifying only two categories, i.e, normal and attack. All entries pertaining to the vulnerabilities listed in Table 4.1 are mapped to just one class i.e, Attack. Both the attack and normal data were manually labeled, and the dataset has been made publicly available[1].

Table 4.1: Distribution of normal and attack data.

Type of vulnerability		**Packets**
normal		6269
attack	au_rand_flooding	655
	truncated_sco_link_request	340
	duplicated_iocap	299
	truncated_lmp_accepted	274
	invalid_feature_page_execution	250
	feature_response_flooding	216
	invalid_timing_accuracy	211
	lmp_overflow_dm1	159
	lmp_auto_rate_overflow	151
	duplicated_encapsulated_payload	111
	invalid_setup_complete	67

[1]ISOT Laboratory, *BrakTooth Attack Dataset*, 2023, https://onlineacademiccommunity.uvic.ca/isot/datasets/

4.2 Summary

Overall, addressing RQ3, we successfully utilized the InternalBlue framework with a Nexus 5 device equipped with the BCM4339 Bluetooth controller to develop an active sniffer. By recompiling the Nexus 5 Bluetooth stack, we unlocked two new debugging ports. Utilizing these ports, and the InternalBlue framework, we applied patches that encapsulate the LMP packets from incoming traffic into HCI events, which are then sent to the host machine via ADB (Android Debug Bridge). These packets are subsequently dissected using the Bluetooth H4 Broadcom Wireshark plugin to extract the LMP packets. To generate normal data, we allowed the active sniffer to exchange media files with benign devices, while attack traffic was generated using the BrakTooth PoC tool. In the following section, we present our results from applying simple machine learning models such as Random Forest, K-NN, and an artificial neural network to classify these attacks.

Chapter 5

Experimental Results and Discussion

5.1 Evaluation Metrics

Out of several evaluation metrics commonly used to evaluate the performance of intrusion detection models, we chose the following set:

1. Accuracy: This is the proportion of observations that were correctly classified as either benign or anomalous. Accuracy is calculated as the ratio of the number of correct predictions to the total number of predictions.

2. True Positive Rate (TPR): This is the proportion of anomalous observations correctly identified by a model as anomalous. It is calculated as the ratio of the number of true positives (i.e., anomalous observations that are correctly classified as anomalous) to the total number of actual anomalous observations.

3. False Positive Rate (FPR): This is the proportion of benign observations that are incorrectly classified by a model as anomalous. It is calculated as the ratio of the number of false positives (i.e., benign observations that are classified as anomalous) to the total number of benign instances.

4. AUC-ROC: The area under the receive operating characteristic curve (AUC-ROC) is a commonly used evaluation metric. The ROC curve shows the relationship between a model's TPR and FPR at different thresholds. AUC-ROC is a value between 0 and 1, which represents the overall performance of the model.

A value of 1 indicates that the model has perfect performance and can perfectly distinguish between benign and anomalous instances, whereas a value of 0.5 indicates that the model's performance is no better than random guessing.

5.2 Contextual Anomaly Detection in IoT

5.2.1 Evaluation procedure

For the experimental evaluation, we split the transformed ISOT AgeTech dataset into an 80–20 ratio to train the chosen machine-learning models. This was in line with the findings of [66], who indicated that the optimal results are obtained by using 20–30% of the data for testing and the remaining 70–80% for training. Additionally, we also computed the evaluation metrics with sliding windows of 15, 25 and 35 samples to determine the influence of window size on performance.

5.2.2 Evaluation Results

In the following, we address RQ2, and present the performance results obtained using the three different classifiers considered in our study on the ISOT AgeTech dataset.

We used the sliding-window technique to transform the ISOT AgeTech dataset into a new set containing correlation and mutual-information scores. These scores were then used as features to train the selected machine-learning classifiers. Stratified K-fold with K = 10 was used for cross-validation, while grid search was employed for hyperparameter tuning. Further, the results from applying these classifiers at different window sizes (15, 25 and 35) are shown in Tables 5.1–5.3.

Table 5.1: Detection performance using a window of size 15 on the transformed ISOT AgeTech dataset.

Classifier	**Accuracy**	**TPR**	**FPR**	**AUC**
Random For.	85.15%	59.45%	0%	79.72%
k-NN	85.15%	65.57%	4.68%	81.44%
Naive Bayes	80.19%	70.27%	14.06%	78.11%

Table 5.2: Detection performance using a window of size 25 on the transformed ISOT AgeTech dataset.

Classifier	**Accuracy**	**TPR**	**FPR**	**AUC**
Random For.	94.91%	84.21%	0%	92.10%
k-NN	**96.61%**	**89.47%**	**0%**	**94.73%**
Naive Bayes	72.88%	36.84%	10%	63.42%

Table 5.3: Detection performance with a window of size 35 on the transformed ISOT AgeTech dataset.

Classifier	**Accuracy**	**TPR**	**FPR**	**AUC**
Random For.	87.50%	65.0%	0%	82.50%
k-NN	89.28%	70.0%	0%	85.0%
Naive Bayes	67.85%	20.0%	5.55%	57.22%

The obtained results are encouraging, particularly with the best results obtained for k-NN and a sliding window size of 25, which yielded high accuracy, AUC, TPR and low FPR. This underscores the strength of the proposed approach when the dataset includes linear or nonlinear relationships between the sensors. Furthermore, to compare the proposed method to traditional deep-learning approaches, we applied the LSTM (Long Short-Term Memory) and the Simple Recurrent Neural Network models to the ISOT AgeTech dataset and performed hyperparameter tuning with grid search. The results presented in Table 5.4 demonstrate that, for this problem, the proposed method, employing relatively simpler machine learning models, achieves comparable performance to traditional deep-learning techniques.

Table 5.4: Results from applying traditional deep-learning approaches on ISOT AgeTech dataset

Classifier	**Accuracy**	**TPR**	**FPR**	**AUC**
Simple RNN	96.32%	88.63%	0.86%	93.89%
LSTM	94.97%	88.96%	2.81%	93.07%

5.2.3 Discussion

Our analysis of the open smart-home, smart-building and ISOT AgeTech datasets, as presented in Section 3.2.1, revealed the presence of correlations among co-located sensors. This phenomenon can be attributed to various reasons:

- Shared environment: When sensors are co-located, they often share a similar environment. This means that external factors such as temperature fluctuations, humidity or light exposure will impact them simultaneously. For example, if a human subject enters a room, there will be a simultaneous change in the readings of many sensors such as temperature, humidity, CO2, PIR (passive infrared) and more.

- Measurement of related phenomena: Some sensors, though distinct, measure phenomena that are inherently related. A classic example is that of temperature and humidity. As the air's temperature increases, if no additional moisture is added to the air, the relative humidity will decrease. Thus, a change in one could lead to a change in the other.

- Sensor Interference: In some cases, the operation or output of one sensor can influence the readings of another sensor nearby. This is especially true if sensors operate on similar frequencies or if one emits a signal (like infrared) that another sensor can pick up.

This finding allowed us to model relationships between sensors using statistical scores and utilize supervised machine learning for classification. Interestingly, these relationships sometimes might also suggest a physical connection between different rooms in a smart home, presenting a promising avenue for future research. The detection model outlined in Section 3.2 was designed to detect sensor readings that deviate from the typical correlation and mutual-information patterns seen so far in normal data. It is important to note that these relationships among sensors are not always present. In such scenarios, there will not be any correlation or mutual information patterns between the time-series readings of different sensors; as a result, our model will not be efficient in detecting anomalies.

Additionally, as the window size increases on the ISOT AgeTech dataset, we observed an improvement in accuracy and TPR. However, this trend eventually reached a critical point beyond which the accuracy and TPR declined. This could be due

to the increased availability of samples under a window, which in turn causes the correlation and mutual-information scores to become more representative of the underlying trend. As the window size increases beyond the critical point, there is also an increase in the possibility of multiple trends in samples being effectively masked into one by correlation and mutual-information calculations causing a reduction in the performance. This trend using the k-NN algorithm is illustrated in Figure 5.1. Therefore, when using this technique, one needs to be cognizant of the window size and tune it to obtain optimal performance.

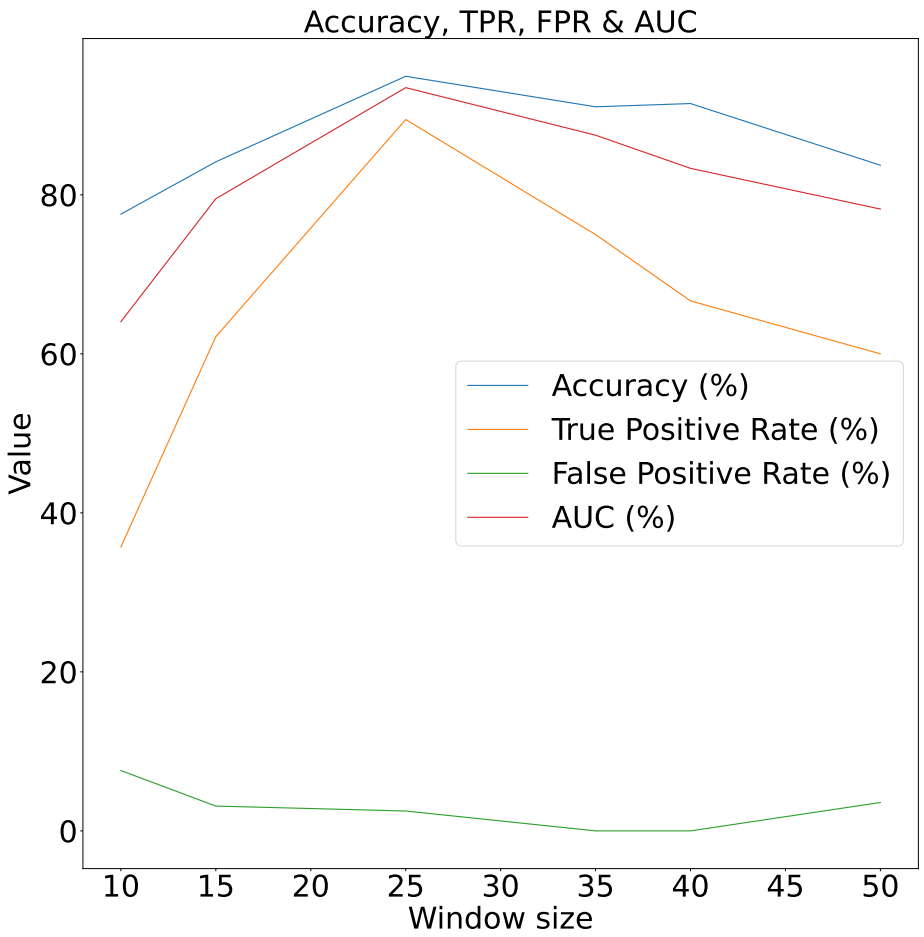

Figure 5.1: Accuracy, TPR, FPR and AUC vs window size of k-NN algorithm on ISOT AgeTech dataset.

As shown in Figure 5.1, when using the k-NN algorithm with the ISOT AgeTech dataset, a window size of 25 provides optimal performance. We also observed that the use of overlapping sliding windows further enhanced performance. Exploring the

effects of overlapping sliding windows and moving averages presents an intriguing avenue for future research.

5.3 BrakTooth Attack Detection

5.3.1 Evaluation Results and Discussion

For the experimental evaluation, we split the dataset into an 80-20 ratio to train three machine learning models: Random Forest, K-Nearest Neighbour, and an Artificial Neural Network. [66] suggested that the best results are obtained if 20-30% of the data is used for testing and the remaining 80% for training. As for evaluation criteria, we chose the true positive rate (TPR), false positive rate (FPR), and precision as they collectively provide a comprehensive understanding of the model's performance in attack detection. TPR captures the model's ability to detect actual attacks, FPR assesses the rate of false alarms, and precision ensures the reliability of positive predictions. We obtained encouraging results, as shown in Table 5.5, with the Random Forest model performing the best, achieving a precision of 99.36% and a true positive rate (TPR) of 98.33%, effectively addressing RQ4. As mentioned in Section 2.2.2, there are no prior works that deal with the problem of detecting BrakTooth-based attacks. Therefore, we will not provide a comparative analysis.

Table 5.5: Results from applying various classifiers

Classifier	**TPR**	**FPR**	**Precision**
Random Forest	98.33%	1.47%	99.36%
ANN	97.05%	7.51%	96.74%
k-NN	97.77%	1.28%	99.43%

Furthermore, due to the lack of non-commercial Bluetooth sniffers, in this paper, we propose an experimental procedure that utilizes an active sniffer to capture attack traffic. However, it is worth noting that in a real-world application, this approach may not be entirely feasible as the attacker would need to specifically target the active sniffer for it to effectively detect any attacks. One strategy to enhance the likelihood of an attack is to periodically change the name of the active sniffer (i.e, victim) to match the names of common devices detected in the Bluetooth piconet being targeted.

Chapter 6

Conclusions and Future Work

6.1 Summary

In the first part of the research, we successfully demonstrated that some co-located sensors in an IoT smart home maintain linear or nonlinear relationships with one another. We then used a sliding window of size S as the contextual attribute under which computations are made for the behavioural attributes, which are the correlation and mutual-information scores among the sensors. These attributes were then successfully used for context-based anomaly detection, achieving high accuracy, decent TPR and low FPR rates. We achieved an accuracy of 96.61%, TPR of 89.47%, FPR of 0% and an AUC of 94.73% with the ISOT AgeTech dataset using a window size of 25 samples. Because we are modelling relationships between sensors using statistical quantities such as correlation and mutual-information scores, it becomes easier for machine-learning classification models to learn and detect any observation that deviates from this relationship, even when dealing with imbalanced distributions of benign and anomalous data. However, our method can only identify anomalies that violate the existing relationships between sensors. This means that intentional or unintentional anomalies that do not violate the relationship between the sensors cannot be detected.

In the second part of our research, we used a two-prong approach to detect BrakTooth-based attacks. First, we set up an attack device using ESP32-WROVER-KIT running a proof of concept (PoC) tool. Second, we set up an active sniffer to capture normal and attack traffic, using the InternalBlue framework and a Broadcom BCM4339 Bluetooth Controller. Finally, we used the collected data to train various

machine-learning models to classify attack data and achieved good performance with the Random Forest model. Our inexpensive and simple detection setup can also be extended to notify various stakeholders, such as caretakers and family members as soon as an attack is detected, thus ensuring the safety and welfare of the senior residents in a smart home. In the future, we intend to enhance our current low-cost setup to capture additional features such as RF signal strength, RF signal frequency offset, bit error rate, invalid data rate, and more to model and identify a broader range of Bluetooth attacks. Further, we also plan to integrate Ubertooth One, to extend the existing setup to detect SweynTooth-based [67] attacks on BLE devices.

6.2 Broader Implications for IoT Security and Future Work

An edge device is a combination of a sensor/actuator and a microcontroller unit that features built-in support for communication modules, including but not limited to WiFi, Bluetooth, and ZigBee. A wide spectrum of these devices are being increasingly adopted in smart homes for a multitude of applications. This evolution of the IoT landscape brings forth pronounced security implications. As highlighted in our research, even established technologies like Bluetooth can have vulnerabilities, emphasizing the interconnected risks in the IoT ecosystem. Attacks can be carefully orchestrated such that compromising a single device could significantly endanger the safety of smart home residents. Additionally, with attackers adapting their techniques continuously, there is an irrefutable need for active monitoring and dynamic defensive measures. In our work focused on active monitoring, we utilized a limited number of sensors, including temperature, humidity and air-quality sensors, to build our detection model. However, as we intend for future work, it would be beneficial to include more sensors, such as luminance sensors to detect ambient lighting, PIR sensors to detect motion, moisture sensors to detect incontinence [68], smart scales to measure weight [69], fall-detection sensors to detect falls [70] and wearable sensors to detect heart rate, sleep and steps taken. This would result in more relationships, which could make our technique more robust. Further, as discussed in 3.2.1, the idea of using the relationships between sensors to predict the physical connection between various rooms in a smart home or a building is an interesting avenue for future work. During our research, we also arrived at a realization about the usefulness of fuzzing

in finding security vulnerabilities. In fact, the BrakTooth vulnerabilities were detecting while fuzzing the LMP layers of the bluetooth stack. Developing fuzzing tools that operate on low-cost hardware and can target a wide variety of IoT devices and communication protocols is a vital research avenue to uncover potentially devastating IoT vulnerabilities.

Bibliography

[1] Patrick Gerland, Sara Hertog, Mark Wheldon, Vladimira Kantorova, Danan Gu, Giulia Gonnella, Ivan Williams, Lubov Zeifman, Guiomar Bay, Helena Castanheira, Yumiko Kamiya, Lina Bassarsky, Victor Gaigbe-Togbe, and Thomas Spoorenberg. *World Population Prospects 2022: Summary of results*. 07 2022.

[2] Vicki Freedman, Linda Martin, and Robert Schoeni. Recent trends in disability and functioning among older adults in the united states: A systematic review. *JAMA : the journal of the American Medical Association*, 288:3137–46, 01 2003.

[3] Gabi Redford. New tech options are helping seniors age in place, Mar 2018.

[4] Becky K White, Annegret Martin, and James White. Gamification and older adults: Opportunities for gamification to support health promotion initiatives for older adults in the context of covid-19. *The Lancet Regional Health–Western Pacific*, 35, 2023.

[5] Peng Liu, Guichen Li, Shengqian Jiang, Yufei Liu, Minmin Leng, Jinping Zhao, Shuo Wang, Xiangfei Meng, Binghan Shang, Li Chen, and Samuel H. Huang. The effect of smart homes on older adults with chronic conditions: A systematic review and meta-analysis. *Geriatric Nursing*, 40(5):522–530, 2019.

[6] Marina Moraitou, Adamantia Pateli, and Sotiris Fotiou. Smart health caring home: A systematic review of smart home care for elders and chronic disease patients. In Panayiotis Vlamos, editor, *GeNeDis 2016*, pages 255–264, Cham, 2017. Springer International Publishing.

[7] Committee on Metabolic Monitoring for Military Field Applications and Institute of Medicine (US). Committee on Military Nutrition Research. *Monitoring Metabolic Status: Predicting Decrements in Physiological and Cognitive Performance*. National Academy Press, 2004.

[8] Nancy J Donovan and Dan Blazer. Social isolation and loneliness in older adults: review and commentary of a national academies report. *The American Journal of Geriatric Psychiatry*, 28(12):1233–1244, 2020.

[9] Alisa Frik, Leysan Nurgalieva, Julia Bernd, Joyce Lee, Florian Schaub, and Serge Egelman. Privacy and security threat models and mitigation strategies of older adults. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*, pages 21–40, Santa Clara, CA, August 2019. USENIX Association.

[10] Ian Zhou, Imran Makhdoom, Negin Shariati, Muhammad Ahmad Raza, Rasool Keshavarz, Justin Lipman, Mehran Abolhasan, and Abbas Jamalipour. Internet of things 2.0: Concepts, applications, and future directions. *IEEE Access*, 9:70961–71012, 2021.

[11] Matheus E Garbelini, Sudipta Chattopadhyay, Vaibhav Bedi, Sumei Sun, and Ernest Kurniawan. Braktooth: Causing havoc on bluetooth link manager, 2021.

[12] Achyuth Nandikotkur, Issa Traore, and Mohammad Mamun. Seniorsentry: Correlation and mutual information-based contextual anomaly detection for aging in place. *Sensors*, 23(15), 2023.

[13] BrakToothPoC. Braktooth proof of concept, 2021. https://github.com/Matheus-Garbelini/braktooth_esp32_bluetooth_classic_attacks.

[14] Achyuth Nandikotkur., Issa Traore., and Mohammad Mamun. Detecting braktooth attacks. In *Proceedings of the 20th International Conference on Security and Cryptography - SECRYPT*, pages 787–792. INSTICC, SciTePress, 2023.

[15] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[16] Andrew A Cook, Göksel Mısırlı, and Zhong Fan. Anomaly detection for iot time-series data: A survey. *IEEE Internet of Things Journal*, 7(7):6481–6494, 2019.

[17] Michael A Hayes and Miriam AM Capretz. Contextual anomaly detection framework for big sensor data. *Journal of Big Data*, 2(1):1–22, 2015.

[18] Chris U Carmona, François-Xavier Aubet, Valentin Flunkert, and Jan Gasthaus. Neural contextual anomaly detection for time series. *arXiv preprint arXiv:2107.07702*, 2021.

[19] Ailin Deng and Bryan Hooi. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4027–4035, 2021.

[20] Georg Ferdinand Schneider, Mads Holten Rasmussen, Peter Bonsma, Jyrki Oraskari, and Pieter Pauwels. Linked building data for modular building information modelling of a smart home. In *eWork and eBusiness in Architecture, Engineering and Construction*, pages 407–414. CRC Press, 2018.

[21] Dezhi Hong, Quanquan Gu, and Kamin Whitehouse. High-dimensional time series clustering via cross-predictability. In *Artificial Intelligence and Statistics*, pages 642–651. PMLR, 2017.

[22] Douglas M Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

[23] Boris Iglewicz and David C Hoaglin. *Volume 16: how to detect and handle outliers*. Quality Press, 1993.

[24] Monowar H Bhuyan, Dhruba Kumar Bhattacharyya, and Jugal K Kalita. Network anomaly detection: methods, systems and tools. *Ieee communications surveys & tutorials*, 16(1):303–336, 2013.

[25] Omar Alghushairy, Raed Alsini, Terence Soule, and Xiaogang Ma. A review of local outlier factor algorithms for outlier detection in big data streams. *Big Data and Cognitive Computing*, 5(1):1, 2020.

[26] Xu Li, Songren Deng, Lifang Li, and Yunchuan Jiang. Outlier detection based on robust mahalanobis distance and its application. *Open Journal of Statistics*, 9(1):15–26, 2019.

[27] Zhenguo Chen and Yong Fei Li. Anomaly detection based on enhanced dbscan algorithm. *Procedia Engineering*, 15:178–182, 2011.

[28] Harjinder Kaur, Gurpreet Singh, and Jaspreet Minhas. A review of machine learning based anomaly detection techniques. *arXiv preprint arXiv:1307.7286*, 2013.

[29] Garazi Artola, Eduardo Carrasco, Kristin May Rebescher, Nekane Larburu, and Idoia Berges. Behavioral anomaly detection system for the wellbeing assessment and lifestyle support of older people at home. *Procedia Computer Science*, 192:2047–2057, 2021.

[30] Zahraa Khais Shahid, Saguna Saguna, and Christer Åhlund. Detecting anomalies in daily activity routines of older persons in single resident smart homes: Proof-of-concept study. *JMIR aging*, 5(2):e28260, 2022.

[31] Oya Aran, Dairazalia Sanchez-Cortes, Minh-Tri Do, and Daniel Gatica-Perez. Anomaly detection in elderly daily behavior in ambient sensing environments. In *Human Behavior Understanding: 7th International Workshop, HBU 2016, Amsterdam, The Netherlands, October 16, 2016, Proceedings 7*, pages 51–67. Springer, 2016.

[32] Chenglong Fu, Qiang Zeng, and Xiaojiang Du. Hawatcher: Semantics aware anomaly detection for appified smart homes. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 4223–4240, 2021.

[33] Han Li, Xinyu Wang, Zhongguo Yang, Sikandar Ali, Ning Tong, and Samad Baseer. Correlation-based anomaly detection method for multi-sensor system. *Computational Intelligence and Neuroscience*, 2022, 2022.

[34] Ece Calikus, Sławomir Nowaczyk, and Onur Dikmen. Context discovery for anomaly detection. 2022.

[35] Erina Ferro and Francesco Potorti. Bluetooth and wi-fi wireless protocols: a survey and a comparison. *IEEE Wireless Communications*, 12(1):12–26, 2005.

[36] Bluetooth SIG. Bluetooth market update. (https://www.bluetooth.com/2023-market-update/), 2023.

[37] Flo Wagner, Jenny Basran, and Vanina Dal Bello-Haas. A review of monitoring technology for use with older adults. *Journal of geriatric physical therapy*, 35(1):28–34, 2012.

[38] Saad El Jaouhari and Eric Bouvet. Secure firmware over-the-air updates for iot: Survey, challenges, and discussions. *Internet of Things*, 18:100508, 2022.

[39] NordicSemiconductor. Device firmware update process, 2018. https://infocenter.nordicsemi.com/topic/com.nordic.infocenter.sdk5.v15.0.0/lib_bootloader_dfu_process.html.

[40] Marco Cominelli, Francesco Gringoli, Paul Patras, Margus Lind, and Guevara Noubir. Even black cats cannot stay hidden in the dark: Full-band deanonymization of bluetooth classic devices. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 534–548, 2020.

[41] Michael Ossmann. Ubertooth one, 2011 https://github.com/greatscottgadgets/ubertooth.

[42] nRFSniffer. Nrf sniffer for bluetooth le, 2023. https://www.nordicsemi.com/Products/Development-tools/nrf-sniffer-for-bluetooth-le.

[43] Tom Nijholt, Erik Poll, and Frits Vaandrager. Bluespec: Development of an lmp state machine and a stateful black-box br/edr lmp fuzzer. 2020.

[44] Shane Ditton, Ali Tekeoglu, Korkut Bekiroglu, and Seshadhri Srinivasan. A proof of concept denial of service attack against bluetooth iot devices. In *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 1–6. IEEE, 2020.

[45] Dennis Mantz, Jiska Classen, Matthias Schulz, and Matthias Hollick. Internalblue-bluetooth binary patching and experimentation framework. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pages 79–90, 2019.

[46] Matthias Schulz, Daniel Wegemer, and Matthias Hollick. Nexmon: A cookbook for firmware modifications on smartphones to enable monitor mode. *arXiv preprint arXiv:1601.07077*, 2015.

[47] Jianliang Wu, Yuhong Nan, Vireshwar Kumar, Mathias Payer, and Dongyan Xu. Blueshield: Detecting spoofing attacks in bluetooth low energy networks. In *RAID*, pages 397–411, 2020.

[48] Daniele Antonioli, Nils Ole Tippenhauer, and Kasper Bonne Rasmussen. The knob is broken: Exploiting low entropy in the encryption key negotiation of bluetooth br/edr. In *USENIX Security Symposium*, pages 1047–1061, 2019.

[49] Terrence OConnor and Douglas Reeves. Bluetooth network-based misuse detection. In *2008 Annual Computer Security Applications Conference (ACSAC)*, pages 377–391. IEEE, 2008.

[50] Yicai Huang, Pengcheng Hong, and Bin Yu. Design of bluetooth dos attacks detection and defense mechanism. In *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, pages 1382–1387. IEEE, 2018.

[51] Pratik Satam, Shalaka Satam, and Salim Hariri. Bluetooth intrusion detection system (bids). In *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, pages 1–7, 2018.

[52] Achyuth Nandikotkur. Aging in place security and privacy, 2021. https://github.com/isotlaboratory/aip-security-and-privacy/.

[53] Concettina Marino, Antonino Nucara, Giorgia Peri, Matilde Pietrafesa, and Gianfranco Rizzo. A generalized model of human body radiative heat exchanges for optimal design of indoor thermal comfort conditions. *Solar Energy*, 176:556–571, 2018.

[54] Sebastian Wilhelm, Dietmar Jakob, and Diane Ahrens. Human presence detection by monitoring the indoor co2 concentration. 2020.

[55] . Wodxgod. Wodxgod/pybot: A simple ddos botnet with basic authentication system written in python, 2018.

[56] Nour Moustafa. A new distributed architecture for evaluating ai-based security systems at the edge: Network ton_iot datasets. *Sustainable Cities and Society*, 72:102994, 2021.

[57] Tim M Booij, Irina Chiscop, Erik Meeuwissen, Nour Moustafa, and Frank TH den Hartog. Ton_iot: The role of heterogeneity and the need for standardization of features and attack types in iot network intrusion data sets. *IEEE Internet of Things Journal*, 9(1):485–496, 2021.

[58] Abdullah Alsaedi, Nour Moustafa, Zahir Tari, Abdun Mahmood, and Adnan Anwar. Ton_iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems. *Ieee Access*, 8:165130–165150, 2020.

[59] Nour Moustafa, Marwa Keshky, Essam Debiez, and Helge Janicke. Federated ton_iot windows datasets for evaluating ai-based security applications. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 848–855. IEEE, 2020.

[60] Nour Moustafa, Mohiuddin Ahmed, and Sherif Ahmed. Data analytics-enabled intrusion detection: Evaluations of ton_iot linux datasets. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 727–735. IEEE, 2020.

[61] Nour Moustafa. A systemic iot–fog–cloud architecture for big-data analytics and cyber security systems: a review of fog computing. *Secure Edge Computing*, pages 41–50, 2021.

[62] Javed Ashraf, Marwa Keshk, Nour Moustafa, Mohamed Abdel-Basset, Hasnat Khurshid, Asim D Bakhshi, and Reham R Mostafa. Iotbot-ids: A novel statistical learning-enabled botnet detection framework for protecting networks of smart cities. *Sustainable Cities and Society*, 72:103041, 2021.

[63] Anuroop Gaddam, Tim Wilkin, Maia Angelova, and Jyotheesh Gaddam. Detecting sensor faults, anomalies and outliers in the internet of things: A survey on the challenges and solutions. *Electronics*, 9(3):511, 2020.

[64] A. Nandikotkur. Isot agetech dataset, 2023. Uploaded on July 2023. https://onlineacademiccommunity.uvic.ca/isot/datasets/.

[65] Natarajan Meghanathan. Assortativity analysis of real-world network graphs based on centrality metrics. *Comput. Inf. Sci.*, 9(3):7–25, 2016.

[66] Afshin Gholamy, Vladik Kreinovich, and Olga Kosheleva. Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation. 2018.

[67] Matheus E Garbelini, Chundong Wang, Sudipta Chattopadhyay, Sumei Sun, and Ernest Kurniawan. Sweyntooth: Unleashing mayhem over bluetooth low energy. In *Proceedings of the 2020 USENIX Conference on Usenix Annual Technical Conference*, pages 911–925, 2020.

[68] Johan Sidén, Andrei Koptioug, and Mikael Gulliksson. The" smart" diaper moisture detection system. In *2004 IEEE MTT-S International Microwave Symposium Digest (IEEE Cat. No. 04CH37535)*, volume 2, pages 659–662. IEEE, 2004.

[69] Simone Casciaro, Lucio Massa, Ilaria Sergi, and Luigi Patrono. A smart pill dispenser to support elderly people in medication adherence. In *2020 5th International Conference on Smart and Sustainable Technologies (SpliTech)*, pages 1–6. IEEE, 2020.

[70] Tao Xu, Yun Zhou, and Jing Zhu. New advances and challenges of fall detection systems: A survey. *Applied Sciences*, 8(3):418, 2018.