HEKA: A Novel Intrusion Detection System for Attacks to Personal Medical Devices

AKM Iqtidar Newaz, Amit Kumar Sikder, Leonardo Babun, and A. Selcuk Uluagac

Cyber-Physical Systems Security Lab
Department of Electrical and Computer Engineering
Florida International University, Miami, USA
{anewa001, asikd003, lbabu002, suluagac}@fiu.edu

***Abstract*—Modern Smart Health Systems (SHS) involve the concept of connected personal medical devices. These devices significantly improve the patient's lifestyle as they permit remote monitoring and transmission of health data (i.e., telemedicine), lowering the treatment costs for both the patient and the healthcare providers. Although specific SHS communication standards (i.e., ISO/IEEE 11073) enable real-time plug-and-play interoperability and communication between different personal medical devices, they do not specify any features for secure communications. In this paper, we demonstrate how personal medical device communication is indeed vulnerable to different cyber attacks. Specifically, we show how an external attacker can hook into the personal medical device's communication and eavesdrop the sensitive health data traffic, and implement man-in-the-middle, replay, false data injection, and denial-of-service attacks. Furthermore, we also propose an Intrusion Detection System (IDS), HEKA, to monitor personal medical device traffic and detect attacks on them. HEKA passively hooks into the personal medical traffic generated by medical devices to learn the contiguous sequence of packets information from the captured traffic and detects irregular traffic-flow patterns using an n-gram-based approach and different machine learning techniques. We implemented HEKA in a testbed consisting of eight off-the-shelf personal medical devices and evaluated its performance against four different attacks. Our extensive evaluation shows that HEKA can effectively detect different attacks on personal medical devices with an accuracy of 98.4% and F1-score of 98%.**

***Index Terms*—Smart health system, Personal medical device, Cyber attacks, Intrusion detection, Medical security**

I. INTRODUCTION

Modern Smart Health Systems (SHSs) significantly increase the efficacy of patients' treatment while reducing healthcare costs for both patients and healthcare providers. In this ecosystem, Personal Medical Devices (PMDs) play a key role in the SHS's success. The PMDs are highly interconnected entities capable of performing traditional healthcare operations while enabling remote monitoring and transmission of health data. Indeed, the global PMD market is expected to reach U.S. $ 35.6 Billion by 2024, registering a Compound Annual Growth Rate (CAGR) of 8.6% during 2019-2024 [1].

As several different manufacturers and technologies compete for a share into the PMD's market, the integration of diverse PMDs into a common healthcare ecosystem can be challenging. As a result, the ISO/IEEE 11073 medical standard [2] has been developed to provide real-time plug-and-play interoperability and communications between PMDs and external systems. As the ISO/IEEE 11073 standard is transport layer agnostic, it is supported by almost any packet-based technology such as TCP/IP, Bluetooth Low Energy (BLE), and Zigbee. However, the ISO/IEEE 11073 standard does not provide any security features for healthcare data exchange and patient monitoring [3]. Instead, it fully relies on the transport layer implementations of already-known protocols to secure the communications. In fact, an attacker capable of exploiting well-known vulnerabilities in communication protocols like BLE or Wi-Fi could gain access to sensitive healthcare data from PMDs devices [4]–[6]. Similarly, in recent years, security concerns for modern SHSs have been rising in both the healthcare sector and academia. Researchers demonstrated that cyber attacks against commercially available PMDs, including attack scenarios like remotely disabling and reprogramming the behavior of the PMDs, are possible [7]. However, the majority of the studied attacks are device-specific and only apply for a specific type of PMDs. As developers and vendors are adapting general standards for interoperability in PMDs, more generalized approaches are needed to understand the security vulnerabilities of the diverse PMDs properly.

In this paper, we first demonstrate that different types of PMDs are vulnerable to cyber attacks by implementing five different attacks on commercially available real-life medical devices. We show how an external attacker can gain access to PMD traffic simply using publicly available tools [8] and software [9] to eavesdrop sensitive patient information, while also disrupting the PMD's communication via Denial-of-Service (DoS), Man-in-the-Middle (MITM), replay, and false data injection attacks. Further, we design and implement an intrusion detection system, HEKA, specifically tailored to monitor PMD's traffic and detect cyber attacks. First, HEKA passively hooks into the PMD's communication and generates different size n-grams from different traffic features such as PDU types, sequential traffic patterns, etc. Then, these features are fed to different Machine Learning (ML) techniques to detect irregular traffic-flow patterns in PMD communication. To test the efficacy of HEKA, we built a PMD communication testbed consisting of 8 off-the-shelf medical devices and implemented HEKA to monitor PMD traffic and detect malicious events using four different ML techniques. Furthermore, we evaluated the performance of HEKA against four different cyber attacks. Our evaluation shows that HEKA can detect malicious activities in PMD's communication with an average accuracy of 98.4% and F1-score of 98%.

Contributions: Our contributions are three-fold:

978-1-7281-4760-4/20/$31.00 ©2020 IEEE

- We effectively performed five different attacks to four different types (eight in total) of commercially available PMDs and uncovered vulnerabilities of real PMDs.
- We proposed HEKA, an Intrusion Detection System to identify different cyber attacks on real-life PMDs. We generated multiple-size n-grams by combining different PMD traffic features (e.g., PDU types, PDU patterns, etc.) and train different ML models using these features to detect malicious network events on PMDs.
- We built a testbed with eight different PMDs to implement and test our proposed framework against four different types of attacks. Our evaluation results from different ML algorithms demonstrate that HEKA is very effective in detecting different threats to PMDs with high accuracy and F1-score.

Organization: The rest of the paper is organized as follows: We provide an overview of security vulnerabilities in PMDs and existing solutions in Section II. The detailed overview of the communication architecture of PMDs in Section III. In Section IV, we discuss our attack environment and how we perform our attacks on PMDs. In Section V, the detailed overview of HEKA is provided. We illustrate the efficiency of HEKA in detecting several malicious activities by analyzing several performance metrics in Section VI. We discuss the benefits of HEKA and future work in Section VII. Finally, we conclude the paper in Section VIII.

II. Related Work

In this section, we discuss different attacks on PMDs and explain the shortcomings of existing security solutions available for PMDs.

A. Existing Attacks

The latest advancements in healthcare with the incorporation of smart health systems and the introduction of PMDs are promising, but at the same time, they introduce unforeseen security risks to healthcare organizations and patients under their care. Recent works have reported several threats to smart health systems, including PMDs. These threats target either the implementation flaws in communication protocols [10]–[13] or device-specific vulnerabilities [14], [15] to perform malicious activities in PMDs. Wood et al. [16] proposed a method to capture network traffic from PMDs and detected plain text transmission of the packet payload that could leak sensitive medical information. Classen et al. [17] analyzed the entire Fitbit ecosystem and combine different approaches such as protocol analysis, software decompiling, static, and dynamic embedded code analysis to reverse engineer the communication protocol used in the device. Here, researchers have been able to get all recently recorded fitness measurement data, inject malicious firmware, and modify the associated smartphone app to disable supported security mechanisms (i.e., authentication and encryption). In another work, a group of researchers illustrated how an attacker could possibly intercept and modify medical and patient data before it has stored in the cloud [18]. Li et al. performed both passive (eavesdropping) and active attacks (impersonation and control of the medical devices to alter the intended therapy) on diabetes therapy systems using public-domain information and widely available off-the-shelf hardware [19]. Halperin et al. reported a potential DoS threat on battery-powered implantable cardioverter-defibrillators (ICDs) by reverse-engineering the communication protocol and performing software radio attacks [7].

B. Existing Solutions

Although researchers and developers have reported several attacks on PMDs, there are no comprehensive security solutions for PMDs. Most of the proposed solutions are device-specific [17] or attack-specific [20], which can not address the security needs of PMDs in a holistic way. Li et al. [19] proposed three possible defense mechanisms based on rolling codes, body-coupled communication, wireless communication monitoring, and anomaly detection to mitigate the security risks associated with medical devices significantly. Siddiqi et al. [21] developed a scheme to ensure the reliability of the timestamp processes in medical data that operated within the resource-limited wearable devices, and accommodated variable latencies over Internet paths. Oconner et al. [22] presented a Bluetooth intrusion detection system to identify reconnaissance, DoS, and information theft attacks on Bluetooth enabled devices, using signatures of the attack. In a recent work, a group of researchers proposed an anomaly-based intrusion detection system for Bluetooth networks that used n-gram based approach to characterize the normal behavior of the protocol [23].

C. Difference with Existing Attacks and Solutions

The main differences between the prior works and our work are as follows: (1) while existing attacks mostly performed MITM or replay attacks [12], [14]–[16], [18], we perform five different attacks on BLE-based PMDs; (2) unlike attacks that targeted specific devices [11], [17], [18], [24], we perform attacks on four different types of real-life PMDs illustrating broader impact of our proposed attacks. In addition, the intrusion detection system that we are proposing constitutes a novel approach to detect malicious attacks in PMDs. Specifically, our approach aims to identify four different types of attacks for PMDs targeting not only specific fitness tracker devices, but commodity PMDs. Although other works propose IDS for Wi-Fi and Bluetooth enabled devices [20], [22], [23], we introduce a novel intrusion detection system for medical devices that use BLE as the targeted communication protocol.

III. Background

In this section, we discuss how a PMD communicates with a manager (smartphone) following the ISO/IEEE 11073 standard [2] and the underlying transport layer protocol stack.

A. PMD Communication Architecture

The ISO/IEEE 11073 medical standard is used to define and monitor the real-time communication processes between PMDs and the controlling applications (Figure 1). PMD's communication involves two main entities: the *agent* and the *manager*. In the example shown in Figure 1, the agent (i.e., PMD device) features an A&D blood pressure monitor [25] that measures the blood pressure and heart rate of the patient. On the other side of the communication channel, the manager is represented by a smartphone, which is assumed to have a higher availability of computing resources if compared with the PMD device. ISO/IEEE 11073 defines an asymmetric communication

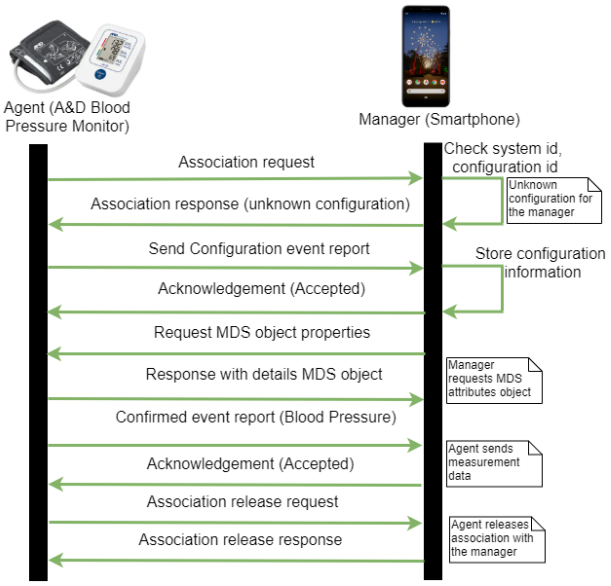


Fig. 1: An example of data exchange diagram between a PMD and its manager.

principle where agents can communicate with only one single manager at any time. In contrast, managers are normally capable of communicating with multiple agents simultaneously. Right after the blood pressure monitor is turned on, it sends an *association request* to the corresponding manager. In cases where this is the first time the agent is requesting association to the manager, the agent also has to send a *configuration report* containing details of all the healthcare-related objects and their static attributes that the agent is capable of handling (e.g., systolic pressure, diastolic pressure, and pulse rate). The manager may also request the *Medical Device System (MDS) object*, which contain host-specific information, unique for every PMD. After receiving the requested information, the manager stores the PMD data as a *device profile* for future communication requests, and sends a *confirmation event report* back to the agent. Then, this report has to be acknowledged by the agent with updated values of the healthcare-related data. Finally, once the communication and data transfer is completed, the agent sends an *association release request* to the manager to release the connection.

B. Communication Protocol Stack

We discuss the Health Device Profile (HDP) for BLE-based communications as the PMDs included in our testbed implement BLE communication. Figure 2 details the elements that form a BLE HDP. The medical application features the actual device application used to control the PMD, which contains the User Interface (UI) used to convert the medical data into a human-readable representation and provides integration to the ISO/IEEE 11073 standard. The ISO/IEEE 11073 standard stack performs building, transmission, and parsing of the IEEE format Protocol Data Unit (PDU) for the agent/manager association and directly links to the HDP. The HDP is the core BLE profile designed to facilitate the transmission and reception of medical device data among agents (PMDs) and managers (recipient of the information). Also, the Generic Access Profile

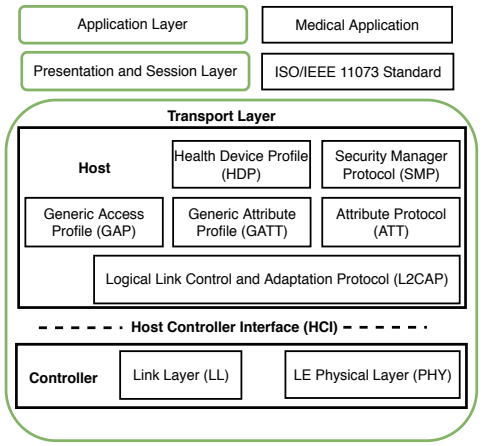


Fig. 2: Health device profile for BLE-based communications between medical agents and managers as defined by the ISO/IEEE 11073 standard.

(GAP) specifies the roles, modes, and procedures of a PMD device. Besides, it manages the connection and advertising procedures of the agent device. The Generic Attribute Profile (GATT) is used by the HDP to define how device data is stored and exchanged between BLE-based devices utilizing the principle called *services and characteristics*. *Service* breaks the device data in logic entities that contain a specific portion of medical data called *characteristics*. The Attribute Protocol (ATT) transfers the attribute data between the agent and the manager while the Security Manager Protocol (SMP) manages the pairing procedure among BLE-based devices (PMD and the smartphone for instance), such as exchange of pairing information, authentication, and distribution key to encrypt and decrypt the transferred packets containing the medical data. Specifically, the authentication step follows three main mechanisms to connect with the manager: (1) just works, (2) passkey entry, and (3) out of band method. All of the PMDs included in this research use the ”just works” pairing method, which is a very simple authentication mechanism that has been proven to be vulnerable to brute force attacks and eavesdrop on the connection. Likewise, this method also offers no way of verifying the devices taking part in the connection, and thus it offers no MITM protection. Finally, the Logical Link Control Adaptation Protocol (L2CAP) encapsulates the data from the BLE-higher communication layers into the standard BLE packet format for transmission.

IV. Attack Model

In this section, we explain the attacker's goals and the methodology to perform different attacks on real-life PMDs.

A. Attacker Goals

This work assumes an attacker that attempts to intercept the communication of PMDs in a way that allows her to perform different active and passive malicious attacks such as eavesdropping, false data injection, etc. We categorize the goals of this attacker in the following three categories based on the impact of the attacks on normal PMD's operations:

- **Connection Delay:** An attacker tries to connect with the PMD using a malicious app installed in the manager (smartphone/laptop), and make the device unavailable for an autho-

rized app. For instance, an unknown mobile app scans and connects to an available pulse oximeter while the authorized app fails to find the targeted medical device in the device list as the device is no longer advertising for connection.

- **Data Interception:** An attacker sniffs the PMDs' communications to eavesdrop and collect sensitive information such as the patient's vitals and device information. For example, while a blood pressure monitor connected to an associated mobile app is sending measured blood pressure data, the attacker can capture the communication packets using a sniffer to extract the device and patient-related information (e.g., device model, firmware version, mac address, systolic and diastolic pressure, etc.). This information can be used to intercept communication and initiate different attacks such as replay attacks, false data injection, etc.
- **Data Modification:** An attacker attempts to modify the patient's vitals measured by a PMD to perform malicious activities such as triggering false alerts, altering treatments, etc. For example, the attacker targets to alter the measured value of a smart insulin pump to change the dose administrated to the patient. The attacker performs this attack by intercepting and modifying the communication packet between the pump and the mobile app.

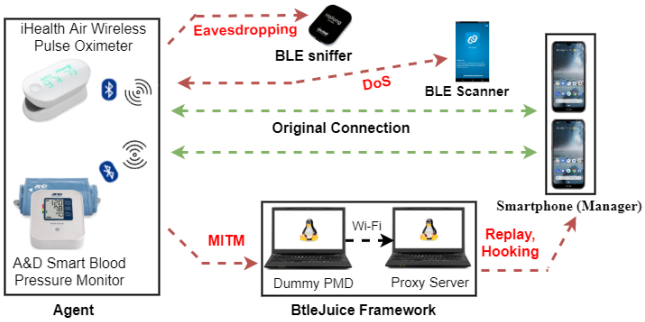


Fig. 3: Our attack environment for PMDs.

B. Attack Environment

We consider the following capabilities for an attacker to successfully implement different attacks on PMDs.

- An attacker has the knowledge of which communication standard and protocol are used by the PMD to establish communication with the manager.
- An attacker has passive access to the communication channel using third-party devices (e.g., sniffer, rogue scanners, etc.).

Figure 3 illustrates the overall attack environment considered in this work. We assume a medical environment consisting of 8 PMDs (e.g., blood pressure monitor, pulse oximeter, weight scales, etc.) on which the attacker implements different attacks. These devices measure different vitals of the patients and communicate with a smartphone (manager) via associated mobile apps. The communication between the PMDs (agent) and the smartphone (manager) uses IEEE 11073 as communication standard and Bluetooth 4.0 (BLE) as the communication protocol. We consider BLE as the devices included in our medical environment use this specific protocol to communicate with the manager. However, the attacks and security solutions proposed in this work can be easily extended to other communication protocols. Specifically, we used three specific tools to set up the attack environment: (1) an unauthorized BLE scanner installed in a rogue device (smartphone), (2) a BLE sniffer to capture the BLE traffic passively, and (3) the *BtleJuice* framework [9]. For the BLE scanner, we used the nRFConnect app on an Android smartphone, which is a universal scanner to discover and connect BLE devices in close proximity [8]. To capture the BLE traffic, we used the Hollong BLE sniffer [26], which can capture the complete set of BLE communication packets between an agent and a manager. Lastly, we used *BtleJuice*, which is a free, publicly available framework to perform Man-in-the-Middle (MITM) attack on BLE devices. To set up this framework, we used two laptops running a Linux-based operating system (Ubuntu 16.04) and BLE adapters. Using the first laptop and a BLE adapter, BtleJuice creates a dummy PMD device that further establishes a connection with real-life PMDs. The second laptop builds a proxy server and connects with the dummy PMD device to create a bridge connection. Finally, using a BLE adapter, the proxy server connects with the smartphone that runs the authorized apps for real-life PMD devices. As the communication between the PMD devices and the manager passes through the dummy device and the proxy server, BtleJuice framework is able to capture the end-to-end communication and perform malicious attacks.

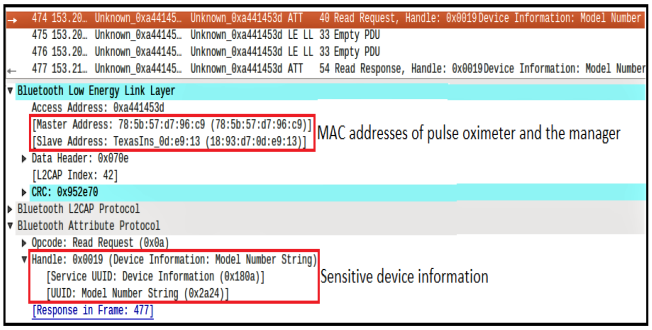
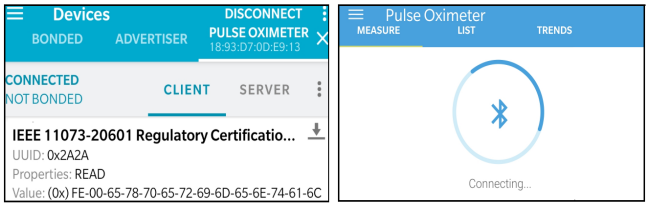


Fig. 4: Captured device information after performing eavesdropping attack.

C. Attack Methodology

In this work, we performed five different types of attacks to actual commodity PMDs. The following discussion explains our attack methodology for each attack. For illustration purposes, we use i-health Pulse Oximeter [27] as our targeted PMD.

Eavesdropping: The main goal of the eavesdropping attack is to passively capture the network traffic between the PMD (i.e., pulse oximeter) and the smartphone without interrupting normal communication. Here, we used a BLE sniffer (Hollong BLE sniffer) to capture the PMD traffic from the pulse oximeter. The captured traffic contained sensitive information such as device information, header information, payload size, etc. For example, in Figure 4, we illustrate a sample of a captured packet between the pulse oximeter and its associated mobile app. By simply analyzing the packet in Wireshark, we can extract the MAC address of the pulse oximeter (18:93:D7:0D:E9:13) and the smartphone (78:5b:57:d7:96:c9), firmware version (214), hardware version (6), and access address (0xa441453d) of the BLE communication. However, we cannot extract any

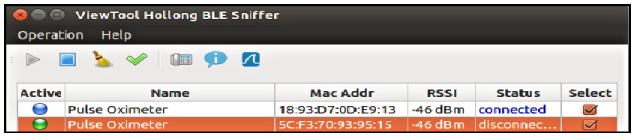


nRF Connect for Mobile App iHealth MyVitals App

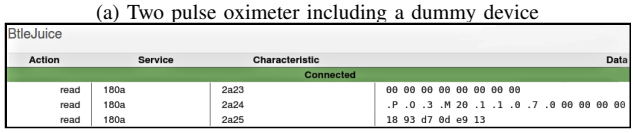
Fig. 5: nRFConnect app denied authorized app to connect with PMD.

payload information in the eavesdropping attack as the payload is encrypted using the BLE link-layer encryption (AES-128 encryption scheme).

DoS attack: The majority of the PMDs use the "Just Work" pairing method, which does not require any authentication process to connect with the associated manager. In a DoS attack, we target this feature to pair an unauthorized app with a targeted PMD. We used nRFConnect, a free mobile app, as an unauthorized app to scan PMDs in close proximity. This unauthorized app scans and connects with any available PMD making the device unavailable to the authorized manager app. Figure 5 shows an example of the DoS attack. Here, a pulse oximeter is connected to the unauthorized app, and a connection request initiated from the authorized manager (iHealth MyVitals app) gets denied. In addition, using the nRFConnect app, we can extract different device information including model number (PO3M 11070), firmware revision string (214), hardware revision string (600), software revision string (108), manufacturer name (iHealth) and its underlying IEEE 11073-20601 regulatory certification number (0xFE006578706572696D656E74616c).



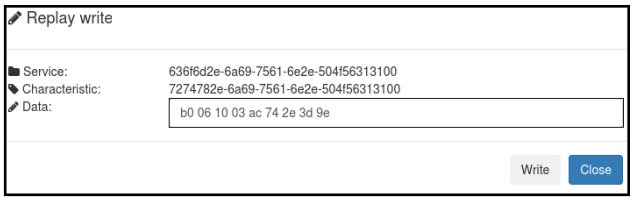
(a) Two pulse oximeter including a dummy device



(b) Captured packet from pulse oximeter

Fig. 6: Captured pulse oximeter packet in proxy server.

Man-in-the-Middle (MITM) attack: To implement the MITM attack, we used the BtleJuice framework to establish a proxy connection between the PMD and the manager. As discussed earlier, BtleJuice uses two Linux-operated machines to act as a dummy device and a proxy server. The targeted PMD which does not perform any authentication process pair with the dummy device running on the first Linux machine via a BLE connection. The second Linux machine initiates a proxy server that is connected with the dummy device via Wi-Fi. Finally, the proxy server acts as a bridge and connects with the manager to complete the connection. Hence, all the communication packets between the PMD and manager reroute through the dummy device and the proxy server. An attacker can observe the BLE packets and extract sensitive information, including device information, payload, etc. forwarded by the PMD. For instance, Figure 6 illustrates a MITM attack on a pulse oximeter. Here, one can observe two pulse oximeters: the real device is connected to the BtleJuice framework, and the dummy device is available for connection. As soon as the dummy device connects with the manager, the BLE sniffer shows only one device disguising the MITM attack from any security tools. One can also see the captured BLE packets in the proxy server in Figure 6.

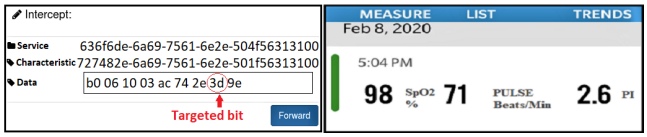


(a) Captured GATT Operation

(b) Same packet captured at manager end

Fig. 7: Write command sent twice to perform a replay attack.

Replay attack: In a replay attack, an attacker aims to send a specific packet in a recurring manner to interrupt normal communication between a PMD and the manager. To implement this attack, we captured the PMD traffic using the BtleJuice framework and select a specific packet to send to the manager app in a loop. This results in the same measured vitals of the patient being shown on the manager's UI, as the connected app is unable to receive new updated patient's health information. Figure 7 illustrates a replay attack in a pulse oximeter. Here, we intercepted a GATT operation between the pulse oximeter and the manager with vendor-specific service and characteristics ID. The payload of the captured GATT operation is "b00610ac742e3d9e" in HexII format, which indicates a write command in the manager's UI - sending this packet in a recurring manner results in same pulse rate being shown on the UI, interrupting the continuous vital update from the pulse oximeter. We can see the multiple occurrences of the same packet on the manager end using Wireshark (Figure 7b).



(a) Modifying payload in packet (b) False pulse rate in manager

Fig. 8: Modified pulse rate showed in the pulse oximeter app.

False data injection: In the false data injection attack, we aimed to change the payload of the captured packet from the PMD. Here, we captured the communication packets from the PMD using the BtleJuice framework and determined the GATT operation, including service and characteristic values. As most of the PMD only sends a numerical value to write on the manager's UI, it is simple to determine the targeted packet containing vital information. We modified the packet manually to change the payload and send it to the manager using the

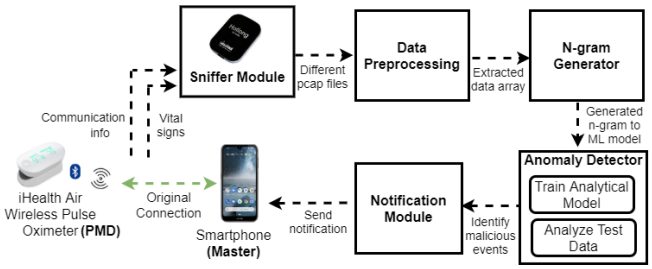


Fig. 9: Our proposed HEKA framework.

proxy server. An example false data injection attack is shown in Figure 8. Here, the captured payload is "b00610ac742e3d9e" in HexII format, where "3d" represents the pulse rate of the patient (46 in decimal format). We manually changed the value to "47" (71 in decimal) and forwarded the modified packet to the manager. In Figure 8b, one can see the modified pulse rate of the patient on the manager's UI.

V. HEKA OVERVIEW

In this section, we present the general architecture of HEKA, a passive intrusion detection system to detect malicious attacks on PMDs. Figure 9 illustrates the overall architecture of HEKA, which includes five main modules: (1) sniffer module, (2) data preprocessing, (3) n-gram generator, (4) anomaly detector, and (5) notification module. The sniffer module captures network traffic between PMDs and manager devices using a sniffer. These collected traffics are preprocessed to remove the noises (e.g., advertise communication and response) in the data preprocessing module and forwarded to the n-gram generator. The n-gram generator uses a sliding window technique to extract sequential patterns in PDU types, which is used as features to detect malicious events in HEKA. The features extracted from the captured traffic are then merged into arrays, so the anomaly detector module can use the feature arrays to train different ML algorithms and builds the detection system. Anomaly detection module also detects whether or not any malicious activities occur within the PMD and the manager communication. Finally, the notification module sends a notification to the manager in the events of an attack detected in the network.

A. Sniffer Module

Sniffer module passively captures PMD traffic from different PMDs without interrupting normal communication. The captured traffic includes control PDUs, data channel PDUs, and empty PDUs. The following equation represents the captured traffic from PMDs: Captured Traffic, $D = \{P_1, P_2, P_3, ..., P_n\}$, where, P_n represents the captured packet at time is the captured packet at time t_n. These captured traffic are forwarded to the data preprocessing module for data sampling and cleaning.

B. Data Pre-processing

Similar to other regular Bluetooth, BLE, or ZigBee devices, PMDs always send advertising packets to connect to a nearby manager device. Additionally, if any nearby manager wants to establish the communication channel, it sends a scan request to the PMDs. Hence, the captured PMD traffic by the sniffer module includes insignificant packets such as advertising packets, scan requests, and scan responses from nearby PMDs. The data preprocessing module collects the PMD traffic captured in the sniffer module and removes the irrelevant packets from the dataset. We only consider complete communication between PMDs and the manager to extract features of benign communication and train HEKA. The data preprocessing module detects complete communication by detecting CONNECT_REQ Protocol Data Unit (PDU) and CONNECT_TER PDU in the dataset. These captured traffic between PMDs and the manager are forwarded to the n-gram generator for feature extraction.

C. N-gram Generator

N-gram generator considers the captured traffic as a contiguous sequence of n items, where items refer to the PDU types of traffic. We consider different sliding windows to extract features from the preprocessed PMD traffic forwarded by the data preprocessing module. We start our sliding window after a CONNECTION_REQ PDU is sent from the manager to the PMD and continue moving it till CONNECTION_TER PDU, which indicates the termination of the communication. We consider 13 different types of PDUs included in the n-gram feature set. Figure 10 shows an example of a 3-gram sequence considered in HEKA. The size of the sliding window in the n-gram generator is configurable and varies from 3 to 6. This configurable feature makes our framework adaptive for different data formats and traffic characteristics. The sequential patterns and the frequencies of PDU types extracted by the n-gram generator are forwarded to the anomaly detector module to train the analytical model and detect malicious events in the PMDs.

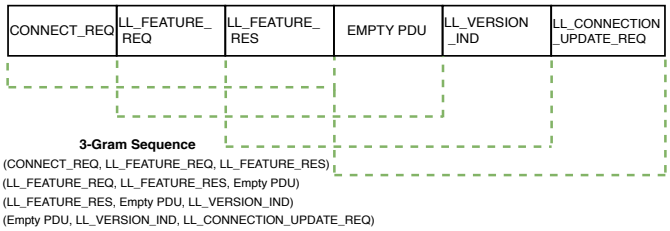


Fig. 10: Our generated 3-gram traffic feature sequence.

D. Anomaly Detector Module

The anomaly detector module uses the generated n-grams from the captured traffic to train different ML algorithms and detect malicious events in the PMD traffic. HEKA aims to detect any malicious events in real-life communication, which requires low detection latency and easy implementation. Based on these needs, we have selected four supervised ML algorithms such as Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and K-Nearest Neighbors (KNN) algorithms to integrate into our framework, as these offers fast computation and easy implementation feature [28]. We briefly discuss these ML algorithms and our rationale to choose them below:

- *Multi-class Support Vector Machine (SVM):* SVM is a supervised machine learning algorithm that maps data to a high-dimensional feature. In contrast, Multi-class SVM is a classification model that can be implemented by converting single class SVM into multiples of the binary classifications [29]. We chose Multi-class SVM because of non-linearity in our dataset and numerical values for our dependent variables (traffic state: benign, MITM, replay, and false data injection).

TABLE I: Our considered feature set for HEKA.

Feature	Description
CONNECT_REQ	Establishment of the connection
LL_FEATURE_REQ	Features set up by the manager's Link Layer
LL_CHANNEL_MAP	Check Link Layer State for any pending real time control
LL_FEATURE_RSP	Feature response from the agent
Empty PDU	Packet acknowledgement, if the agent device has a value
LL_VERSION_IND	Contain company identifier of the manufacturer of the BLE Controller
LL_CONECTION_UPDATE_REQ	Check the state of Link Layer to avoid any rejection from BLE stack
Read By Type Request	manager sends the Read By Type Request to obtain the attribute handler
Read By Type Response	Contain attribute data handles, characteristic handles and UUID's:
Write By Type Request	Request the server to write the value of an attribute
Write By Type Response	Write Request works
Handle Value Notification	Send only when characteristics value changed in the packet
CONNECT_TER	Terminate connection

We use the Radial Basis Function (RBF) as our kernel space because of the minimum number of feature sets.

- *Decision Tree (DT):* DT uses a graph branching method to explain every possible outcome of a decision. It uses the *divide and conquer* approach, and recursively selects the attribute that is used to partition the training dataset into subsets until each leaf node in the tree has a uniform class membership [30]. We consider the frequency pattern of our feature set. For every dependent variable, the frequency pattern of our independent variables (CONNECTION_REQ, LL_FEATURE_REQ, etc.) is different from each other. Different frequency pattern helps the decision tree to find important features in the dataset.
- *Random Forest (RF):* RF is an ensemble learning method consisting of many decision trees to model the classifier. Here, a different subset of training data is selected with a replacement to train each tree [31]. RF model is also effective for estimating missing data. As our framework uses a sniffer to collect the data, we chose RF as it maintains accuracy to mitigate the possibility of missing packets in the communication.
- *K-Nearest Neighbor (KNN):* The KNN algorithm is an instance-based learning algorithm that stores the training samples but does not generate a specific classification model [32]. During classification, distances between test and training samples are calculated, and the test sample is assigned the same class label as its nearest neighbor. KNN offers high accuracy and the faster creation of a raining model for unknown traffic data, which is suitable for HEKA.

E. Notification Module

The notification module of HEKA notifies the manager in the event of any malicious attack against PMDs.

VI. PERFORMANCE EVALUATION

In this section, we evaluate the effectiveness and feasibility of HEKA in detecting malicious activities in PMDs. We consider several research questions to evaluate HEKA in detecting malicious activities.

- **RQ1** What is the performance of HEKA in detecting a single attack to the PMDs? (Sec. VI-C)
- **RQ2** What is the performance of HEKA in detecting combined attacks at once to the PMDs? (Sec. VI-D)
- **RQ3** What is the impact of different n-grams on the performance of the HEKA? (Sec. VI-E)

A. Training Environment and Methodology

To test the efficacy of HEKA, we collected data from four different types (eight in total) of PMDs (iHealth Air Wireless Pulse Oximeter [27], A&D blood pressure monitor [25], QuardioArm blood pressure monitor [33], A&D wireless weight scale [34]) while measuring patient's vitals such as pulse rate, blood pressure, blood oxygen, and weight monitoring. Note that we only collected data from normal users in a lab environment, and no medical and personal data has been recorded other than the captured traffic. While collecting data from the PMDs, we considered variable data collection time as the data stream, and sampling time varies from device to device. For example, pulse oximeter sends a continuous data stream to the manager, while weight scale measures discrete values. Again, some PMDs (e.g., blood pressure monitor) need initialization time to measure the patient's vital properly. To capture the complete PMD traffic properly, we considered three different time windows (10 minutes, 5 minutes, and 2 minutes). We consider the PMD traffic captured while measuring the vitals as benign data. For collecting the malicious data, we performed four different attacks (MITM, replay, false data injection, and DoS) against PMDs, as described in Section IV. Here, we did not consider any passive attacks (e.g., eavesdropping) as passive attacks do not affect the normal communication between PMDs and the manager [35], [36]. We specified the MAC address of the PMDs while capturing PMD traffic in the BLE sniffer to avoid irrelevant packet capture from nearby BLE devices. We used Wireshark and t-shark, two well-known network packet analyzer to pre-process and analyze our captured traffic. As we aimed to detect malicious events on real-life communication, we reduced the feature space to achieve low processing and detection time [37]. Upon analyzing the captured packet, we considered 13 different PDU types as a selected feature for our IDS (illustrated in Table I). We used different n-grams (3 to 6 gram) to extract sequential patterns on selected PDUs. We calculated the frequency of the generated n-grams in the complete packet capture and then considered the top twenty frequency patterns in that traffic flow as features. Finally, we used these features to train different ML models and detect malicious activities on PMDs. We captured 1039 instances of communication consisting of total number packets transmitted between the PMDs and managers, where 731 instances were for benign communication, and 308 instances were captured while performing malicious attacks. We also note that we used 70% of the collected benign traffic to train the ML-models and the remaining 30% of the data along with malicious dataset to test the IDS, which is a common practice [38], [39].

B. Performance Metrics

In the evaluation of HEKA, we used four different performance metrics: Accuracy, Precision, Recall, F1-score. Accuracy refers to the degree of closeness of a measured quality to that quality's true value, and precision calculates the fraction of correct positive identifications. Recall identifies the portion of correctly identified positives while F_1-score measures a test's accuracy considering both precision and recall.

C. Performance of HEKA Against Individual Attacks

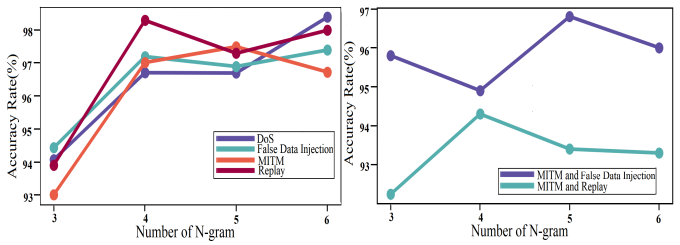
We evaluate the effectiveness of HEKA against the different types of attacks implemented in Section IV. To do so, we performed every single attack separately against the PMDs included in our testbed. Table II shows performance metrics

TABLE II: Performance of different ML-based single attack detection techniques using 4-gram sequence in HEKA.

Attack	MITM				Replay				False data injection				DoS			
Algorithm	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
SVM	.971	.97	.97	.97	.983	.98	.98	.97	.972	.97	.97	.97	.967	.97	.97	.97
KNN	.93	.93	.93	.93	.939	.94	.94	.94	.944	.94	.93	.93	.941	.929	.94	.94
DT	.975	.98	.98	.98	97.3	.97	.97	.97	.969	.97	.96	.97	.967	.97	.97	.97
RF	.967	.96	.97	.97	.98	.98	.98	.98	.974	.97	.97	.97	.984	.98	.98	.97

TABLE III: Performance of HEKA in detecting multiple attacks at the same time.

Attacks	MITM and False data injection				MITM and Replay			
Algorithm	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
SVM	.964	.96	.96	.96	.922	.91	.92	.92
KNN	.937	.94	.94	.94	.903	.90	.90	.90
DT	.957	.96	.96	.96	.922	.92	.92	.92
RF	.949	.95	.95	.95	.943	.94	.94	.94



(a) Individual attack (b) Combined attacks

Fig. 11: Impact of different N-grams sizes in HEKA.

results after evaluating HEKA using the 4-gram approach for different attacks. One can observe that HEKA achieves the highest accuracy of 98.4% and 97.4% using RF algorithms for the case of DoS and false data injection attacks, respectively. Also, SVM achieves the highest accuracy of 98.3% and F1-score of 98% for the case of replay type of attacks. For MITM attack detection, DT algorithms provided the better accuracy results of 97.5%. KNN achieved high false positive and negative rates for MITM and replay attacks because few PMD sent several similar packets to the manager to confirm that the manager receives the patients' vital accordingly. Finally, it is worth noting that KNN achieved the lowest performance results for all the considered attacks.

D. Performance of HEKA Against Combined Attacks

While using the BtleJuice framework, it is possible for an attacker to perform more than one attack on the PMDs simultaneously. Specifically, we found that attackers may be able to combine MITM-False data injection and MITM-Replay types of attacks. To perform both combined attacks, the PMD has to connect first to the core device via the BLE connection. In contrast, the associated smartphone app connects to the interception proxy featured by the BtleJuice framework. We again tested the performance of HEKA against combined attacks while using the 4-gram approach. From Table III, one can observe that SVM performs with the highest accuracy of 96.4% against MITM-False data injection attacks. On the other hand, RF achieved 94.3% accuracy while identifying MITM-Replay type of combined attacks. Also, one can observe that the general performance of HEKA degrades when trying to identify anomaly behaviors of BLE-based PMD communications as a result of combined attacks. Finally, as in the case of single attacks, KNN achieved the lowest performance results.

E. Performance of HEKA with Different N-gram Sequences

We tested the impact of n-gram sizes on the HEKA performance. To do so, we performed anomaly BLE traffic detection while implementing HEKA using four different n-gram sizes: 3-gram, 4-gram, 5-gram, and 6-gram. We performed n-gram size analysis using the RF algorithm only, as it appears to be the one achieving better results on average for the four different single attack cases and the two combined attacks analyzed. For individual attacks, HEKA achieved the lowest accuracy for a 3-gram sequence pattern when identifying the MITM attack (Figure 11a). For the 4-gram sequence, it achieved the highest accuracy of 98.4%. Also, for the case of the 6-gram sequence, HEKA can detect DoS attack with 98% accuracy. For the instance of combined attacks, the HEKA framework obtained the lowest accuracy of 92% for a 3-gram sequence when identifying MITM-Replay type of combined attacks. In contrast, it achieved 96% accuracy for the case of MITM-False data injection attack identification. Finally, both types of attacks obtained degraded performance for the 6-gram sequence, if compared with the 5-gram case (Figure 11b).

VII. DISCUSSION

In this section, we illustrate how HEKA can be effective in real-time malicious activity detection in PMDs.

Scalability- HEKA uses PMD traffic patterns to extract PDU types and detect malicious activities in the network. For a similar type of PMDs, PMD traffic patterns remain almost the same. Hence, HEKA can detect malicious activities in different PMDs without retraining. For connected SHS where multiple PMDs share the same ecosystem, HEKA can be an efficient solution in terms of scalability. Additionally, HEKA uses traffic patterns as a feature to detect malicious activities which can be easily adapted by new communication standard and protocols.

Effectiveness- HEKA achieves high accuracy in detecting different types of malicious attacks on PMDs. One interesting observation is that different ML models achieve high accuracy for different types of attacks in HEKA. As HEKA extracts features from PMD's traffic patterns to detect malicious activities, it is expected that different traffic patterns (caused by the type of attack being performed) will have an impact on the machine learning algorithm achieving the highest accuracy. To avoid tailoring HEKA model to specific attacks, modern ML approaches like AutoML could be a viable solution to automatically select the best ML algorithm for HEKA and ensure the maximum effectiveness for diverse attacks in real-life implementation [39].

Passive analysis- HEKA uses sniffing techniques to passively observe the network traffic between PMDs and the manager without obstructing the normal communication to detect malicious activities. As there is no need to implement HEKA on PMDs or the manager, it is very efficient in terms of performance overhead.

VIII. Conclusions

Personal medical devices (PMDs) offer remote monitoring and automated treatments for the patients improving patient's quality-of-life while lowering the cost. However, communication standards and protocols used by these PMDs raise many security concerns, and attackers can perform different attacks to compromise sensitive information. In this paper, we illustrated different vulnerabilities of PMDs by performing five different types of cyber attacks on commercially available healthcare devices. These attacks demonstrate how an attacker can compromise the communication system of PMDs and perform malicious activities such as denial-of-service, false data injection, etc. Additionally, we presented HEKA, a passive IDS to monitor and detect cyber attacks on the network traffic of PMDs. HEKA passively hooks to the PMDs communication and detects malicious activities by (1) generating different sizes of n-gram using the sequential patterns in PDU types and (2) training different ML models to differentiate benign and malicious communication traffic. We evaluated HEKA on different real-life PMDs and against different cyber attacks. Our evaluation showed that HEKA is highly effective and efficient in detecting different attacks with over 98% of accuracy.

IX. Acknowledgment

This work is supported by the US National Science Foundation (Awards: NSF-CAREER-CNS-1453647, NSF-CNS-1718116, NSF-1663051) and Florida Center for Cybersecurity's Capacity Building Program. The views expressed are those of the authors only, not of the funding agencies.

References

[1] "Global self-care medical device market," https://www.prnewswire.com/news-releases/global-self-care-medical-devices-market-trends-share-size-growth-opportunity-and-forecast-2019-2024-300962107.html/, November 2019.

[2] Y. Tang, X. Duan, T. Fan, H. Feng, W. Jin, and B. Shi, "The study of the implementation and the extension of the iso/ieee-11073 standards," in *Symposium on Computers and Communications (ISCC)*. IEEE, 2017.

[3] Ó. J. Rubio, J. D. Trigo, Á. Alesanco, L. Serrano, and J. García, "Analysis of iso/ieee 11073 built-in security and its potential ihe-based extensibility," *Journal of biomedical informatics*, pp. 270–285, 2016.

[4] A. K. Sikder, G. Petracca, H. Aksu, T. Jaeger, and A. S. Uluagac, "A survey on sensor-based threats to internet-of-things (iot) devices and applications," *arXiv preprint arXiv:1802.02041*, 2018.

[5] A. K. Sikder, A. Acar, H. Aksu, A. S. Uluagac, K. Akkaya, and M. Conti, "Iot-enabled smart lighting systems for smart cities," in *8th IEEE Annual Computing and Communication Workshop and Conference (CCWC)*, 2018.

[6] J. Choi, A. Anwar, H. Alasmary, J. Spaulding, D. Nyang, and A. Mohaisen, "Iot malware ecosystem in the wild: a glimpse into analysis and exposures," in *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, 2019, pp. 413–418.

[7] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel, "Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses," in *IEEE Symposium on Security and Privacy*, 2008.

[8] "nrfconnect for mobile," https://www.nordicsemi.com/Software-and-tools/Development-Tools/nRF-Connect-for-mobile/.

[9] "Btlejuice framework," https://github.com/DigitalSecurity/btlejuice/.

[10] K. Malasri and L. Wang, "Securing implantable devices for healthcare: Ideas and challenges," *IEEE Communications Magazine*, 2009.

[11] K. Lotfy and M. L. Hale, "Assessing pairing and data exchange mechanism security in the wearable internet of things," in *International Conference on Mobile Services (MS)*. IEEE, 2016, pp. 25–32.

[12] V. Pournaghshband, M. Sarrafzadeh, and P. Reiher, "Securing legacy mobile medical devices," in *International Conference on Wireless Mobile Communication and Healthcare*. Springer, 2012, pp. 163–172.

[13] U. Meteriz, N. F. Yıldıran, J. Kim, and D. Mohaisen, "Understanding the potential risks of sharing elevation information on fitness applications."

[14] J. Radcliffe, "Hacking medical devices for fun and insulin: Breaking the human scada system," in *Black Hat Conference presentation*, 2011.

[15] "R7-2016-07: Multiple vulnerabilities in animas onetouch ping insulin pump," https://blog.rapid7.com/2016/10/04/r7-2016-07-multiple-vulnerabilities-in-animas-onetouch-ping-insulin-pump/, October 2016.

[16] D. Wood, N. Apthorpe, and N. Feamster, "Cleartext data transmissions in consumer iot medical devices," in *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*, 2017, pp. 7–12.

[17] J. Classen, D. Wegemer, P. Patras, T. Spink, and M. Hollick, "Anatomy of a vulnerable fitness tracking system: Dissecting the fitbit cloud, app, and firmware," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 1, pp. 1–24, 2018.

[18] M. Aliasgari, M. Black, and N. Yadav, "Security vulnerabilities in mobile health applications," in *2018 IEEE Conference on Application, Information and Network Security (AINS)*. IEEE, 2018, pp. 21–26.

[19] C. Li, M. Zhang, A. Raghunathan, and N. K. Jha, "Attacking and defending a diabetes therapy system," in *Security and Privacy for Implantable Medical Devices*. Springer, 2014, pp. 175–193.

[20] H. Alipour, Y. B. Nashif, P. Satam, and S. Hariri, "Wireless anomaly detection based on ieee 802.11 behavior analysis," *IEEE transactions on information forensics and security*, pp. 2158–2170, 2015.

[21] M. Siddiqi, V. Sivaraman, and S. Jha, "Timestamp integrity in wearable healthcare devices," in *International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. IEEE, 2016, pp. 1–6.

[22] T. OConnor and D. Reeves, "Bluetooth network-based misuse detection," in *Annual Computer Security Applications Conference*. IEEE, 2008.

[23] P. Satam, S. Satam, and S. Hariri, "Bluetooth intrusion detection system (bids)," in *International Conference on Computer Systems*. IEEE, 2018.

[24] H. Fereidooni, T. Frassetto, M. Miettinen, A.-R. Sadeghi, and M. Conti, "Fitness trackers: fit for health but unfit for security and privacy," in *CHASE*. IEEE, 2017.

[25] "A&d blood pressure monitor," https://medical.andonline.com/product/blood-pressure-monitor/.

[26] "Hollong sniffer," http://www.viewtool.com/index.php/en/22-2016-07-29-02-11-32/205-hollong-bluetooth-4-0-4-1-4-2-ble-sniffer-analyzer-software.

[27] "ihealth pulse oximeter," https://ihealthlabs.com/fitness-devices/wireless-pulse-oximeter/.

[28] A. K. Sikder, H. Aksu, and A. S. Uluagac, "6thsense: A context-aware sensor-based attack detector for smart devices," in *26th {USENIX} Security Symposium ({USENIX} Security 17)*, 2017, pp. 397–414.

[29] A. Mathur and G. M. Foody, "Multiclass and binary svm classification: Implications for training and classification users," *IEEE Geoscience and remote sensing letters*, vol. 5, no. 2, pp. 241–245, 2008.

[30] A. Newaz, A. Sikder, A. Rahman, and A. S. Uluagac, "Healthguard: A machine learning-based security framework for smart healthcare systems," in *Social Networks Analysis, Management and Security*. IEEE, 2019.

[31] A. K. Sikder, H. Aksu, and A. S. Uluagac, "A context-aware framework for detecting sensor-based threats on smart devices," *IEEE Transactions on Mobile Computing*, 2019.

[32] S. Güney and A. Atasoy, "Multiclass classification of n-butanol concentrations with k-nearest neighbor algorithm and support vector machine in an electronic nose," *Sensors and Actuators B: Chemical*, 2012.

[33] "Quardioarm blood pressure," https://www.getqardio.com/qardioarm-blood-pressure-monitor-iphone-android/.

[34] "A&d weight machine," https://medical.andonline.com/product/premium-wireless-weight-scale/uc-352ble?commerce_product=4.

[35] Z. B. Celik, L. Babun, A. K. Sikder, H. Aksu, G. Tan, P. McDaniel, and A. S. Uluagac, "Sensitive information tracking in commodity iot," in *27th USENIX Security Symposium)*, 2018, pp. 1687–1704.

[36] A. K. Sikder, L. Babun, Z. B. Celik, A. Acar, H. Aksu, P. McDaniel, E. Kirda, and A. S. Uluagac, "Multi-user multi-device-aware access control system for smart home," *arXiv preprint arXiv:1911.10186*, 2019.

[37] A. K. Sikder, H. Aksu, and A. S. Uluagac, "Context-aware intrusion detection method for smart devices with sensors," Sep. 17 2019, uS Patent 10,417,413.

[38] A. K. Sikder, L. Babun, H. Aksu, and A. S. Uluagac, "Aegis: a context-aware security framework for smart home systems," in *the 35th Annual Computer Security Applications Conference*, 2019.

[39] Google. (2019, May) Cloud automl. [Online]. Available: https://cloud.google.com/automl/