A Home Intrusion Detection System using Recycled Edge Devices and Machine Learning Algorithm

Daewoo Kwon[1], Jinseok Song[2], Chanho Choi[3], Eun-Kyu Lee[4]*
Department of Information and Telecommunication Engineering
Incheon National University, Incheon, Korea

***Abstract*—This paper proposes a home intrusion detection system that makes the best use of a retired smartphone and an existing Wi-Fi access point. On-board sensors in the smartphone mounted on an entrance door records signals upon unwanted door opening. The access point is reconfigured to serve as a home server and thus it can process sensor data to detect unauthorized access to home by an intruder. Recycling devices enables a home owner to build own security system with no cost as well as helps our society deal with millions of retired devices and waste of computing resources in already-deployed IT devices. In order to improve detection accuracy, this paper proposes a detection method that employs a machine learning algorithm and an analysis technique of time series data. To minimize energy consumption on a battery-powered smartphone, the proposed system utilizes as few sensors as possible and offloads all the computation to the home edge server. We develop a prototype and run experiments to evaluate accuracy performance of the proposed system. Results show that it can detect intrusion with probability of 95% to 100%.**

Keywords*—Security; intrusion detection; edge computing; Internet of Things; recycling

I. Introduction

Home security has been a social problem that severely threats lives and activities of the public. These days, people have installed a smart lock at a front door and a web camera at home to enhance their security levels. However, unauthorized intrusion also uses more intelligent and various ways. For instance, it is reported that many web cameras are vulnerable to attacks like malware and that people do not change default passwords set from a factory [1]. Moreover, these devices only protect home in a passive way. A home owner can see the inside of the house, only when she opens a corresponding mobile application; she may be busy on real intrusion events. Registering a professional home security service is another option for her. But, this require service users to pay fees and sometimes to purchase additional equipment, which does not attract people in an economical sense. Moreover, sensitive data such as activity records at home is stored and managed at a server of the service company, introducing privacy issues.

To tackle the problem, this paper proposes a new intrusion detection system that composes of recycled devices that have not been used for a while and/or already exist at home. More specifically, the system (i) uses a retired smartphone and (ii) reuses a Wi-Fi access point device, which saves an additional cost as well as solves other issues.

With fast advancement of a smartphone technology, people tend to upgrade to new smartphones frequently. A tons of old smartphones are abandoned every year or remain unused at home. Unfortunately, we have not found the right way to take advantage of retired smartphones. It is observed that such smartphones are still with advanced on-board sensors such as acceleration sensors, gyro sensors, and humidity sensors. The proposed system tries to make full use of the sensors. A retired smartphone in the system is mounted on a door, and on-board sensors recognize the door opening, which detects unauthorized accesses to a house by intruders. A Wi-Fi access point is a ubiquitous device installed at almost every houses, or even at every rooms; it is hard to imagine home without wireless connectivity today. Such a device is often with computing capability, and latest ones are equipped with powerful CPUs. However, it is observed that the device operates only as a router delivering network packets. Most computing resources remain unused and wasted. The proposed system tries to reuse surplus computing resources at the existing home device. This paper builds a prototype of a Wi-Fi access point server that processes sensor data transmitted from the retired smartphone as well as provides direct wireless connectivity to the smartphone. Placing the server at home also gives benefits. A local data processing implements an edge computing paradigm [2, 3], which makes it possible to analyze data in real time and thus improves latency performance. An event of home intrusion can be immediately captured and appropriate reaction can be taken quickly. Since data is stored at the local server and under a full control of a home owner, the privacy issue is also resolved. We note that this approach also affect our society in a good way beyond personal advantages. Recycling retired, existing devices enables us to save cost for disposing old devices and to make the best of surplus IT resources, otherwise wasted, on devices deployed widely today.

The proposed intrusion detection system faces technical challenges. First, a smartphone is with cheap on-board sensors mainly because of a cost issue. Therefore, it is hard to expect high accuracy in sensor data. Next, the sensors react to a variety of physical phenomena and generate millions of data accordingly. Out of the flood of big data, the system should be able to identify an event of interest (unauthorized door opening in our system) with high probability and to filter out unrelated noise signals. Last, the smartphone runs on a battery, and thus the system should be designed to minimize energy consumption and to maximize its lifetime. In order to resolve the first and second challenges, this paper develops an intrusion detection method that employs the *k*-nearest neighbors

*Corresponding Author
This research was supported by Incheon National University Research Grant in 2018. Eun-Kyu Lee is the corresponding author.

algorithm, a machine learning algorithm used for classification and regression. Our method also employs the dynamic time warping algorithm to avoid the curse of dimensionality in the *k*-nearest neighbors algorithm. In this way, the proposed method is able to detect an intrusion event in a lightweight manner. For the third challenge, our system tries to utilize as small numbers of on-board sensors as possible. In a prototype, it only turns on two sensors, an accelerometer and a magnetometer in the retired smartphone. In addition, the smartphone offloads all the computation to the home edge server, minimizing its duty to save energy. Since the server is AC-powered, it is safe from unexpected power outage.

While our research makes the best use of the retired smartphone, there are still limitations on it. One example is battery condition of the smartphone. In general, the smartphone is likely to be with a battery of bad health. This implies that it should be recharged frequently, which is a big troublesome. An emerging energy harvesting technology may reduce the battery issue. Turning off unnecessary jobs in the smartphone can be another solution approach. By default, the smartphone runs multiple tasks both in a foreground and a background. Many of them are not directly related to our system. Thus, a proper setup can minimize power consumption on the smartphone and thus extend its lifetime. Finding an optimal configuration becomes an interesting research topic that this paper does not address.

The rest of the paper is composed as follows. Section 2 reviews related works. The proposed intrusion detection system is described in detail with background knowledge in Section 3. Next section develops a prototype of the system, which is followed by experiments and performance evaluation of detection accuracy in Section 5. The last section concludes the paper.

II. Related Works

Wu et al. [4] developed a method that detected events of door opening/closing by using a barometer of a smartphone. The sensor measured fluctuation in air pressure inside a house when a door is opening and closing. However, the method turned to be ineffective if the house has another open window(s). Dissanayake et al. [5] proposed an algorithm that recognized events of multiple doors in an indoor environment using a microphone and sound recording capability on a smartphone. It employed an active sound probing and analysis of the Doppler shift that captured acoustic characteristics telling door states (open/close) via impulse response. Mahler et al. [6] developed SecureHouse that used sensors of a smartphone mounted on a wall near a door. The system captured unique vibration signatures of door events, opening and closing, as well as the rotation of a door. Gong et al. [7] proposed an infrastructure-free door event detection approach that utilized built-in magnetic sensors of a smartphone. A key observation is that magnetic signals change patterns especially when a smartphone passes through a door. Behringer et al. [8] proposed a car security system that used GPS and an accelerometer of a smartphone. It could detect a variety of car-related events such as door opening/closing, engine start, and movement. Unlike these research, our system makes use of more than two sensors at the same time, which helps accuracy of event detection. Moreover, it adopts the concept of edge computing, which is implemented by reusing a Wi-Fi access point device.

Wireless signal data has been also used to recognize door events. Ohara et al. [9] proposed a method to detect events on indoor objects. Their system consisted of a Wi-Fi access point and a special Wi-Fi signal receiver that captured changes on Wi-Fi signal propagation during events and analyzed it to detect door opening. Similarly, Shi et al. [10] developed a system that deployed FM-radio signal receivers inside a room and used the same principle that any changes in an indoor environment affected the propagation of radio waves. Hnat et al. [11] developed Doorjamb, an object tracking scheme. It used many ultrasound sensors on doors that enabled to detect moving object in an indoor environment.

There are a list of research projects that make use of open source hardware platforms. Jabbar et al. [12] developed IoT@HoMe, an IoT based home automation system that monitored home conditions and automatically controlled home appliances. It utilizes a node microcontroller unit (NodeMCU) to implement a microcontroller / an Internet gateway that obtains data from sensors and forwards it to a cloud server. Ozeer et al. [13] designed a framework for Fog-IoT applications. Raspberry Pi devices were extensively used in the framework on which lights are controlled appropriately according to owner's modes. Kaur et al. [14] developed a home automation system using Arduino devices and GSM. These research enables to build cost-effective smart home systems, but scarcely addressed how to use retired smartphones that our system adopts to maximize cost benefit of a home intrusion system.

III. Proposed Intrusion Detection System

A. System Architecture

The proposed home intrusion detection system consists of a retired smartphone, a home edge server, and a user carrying a smartphone as shown in Fig. 1. The retired smartphone, mounted on a door, senses the door opening and closing and transmits sensor records to the home server using its built-in Wi-Fi connectivity. The server, placed inside home, analyzes the received data to determine occurrence of a security event. It also sends an alarm message to the user in the case of event occurrence. We note that the goal of our system is to develop an intrusion detection system at a reasonable cost. To this end, the sensors and the server are developed by recycling devices. The sensors are from a retired smartphone, and the server can be made by an edge device like a home Wi-Fi access point.

B. Operation Scenario

Our system operates in two steps: an initialization phase and an intrusion detection phase.

1) Initialization phase: After mounting the retired smartphone, a user registers it to the edge server so that the server recognizes sensor data from the smartphone as authorized inputs. The on-board sensors should be calibrated properly to guarantee accuracy of sensor data. Since our system uses a machine learning algorithm, training data should be collected and stored at the server initially. To this

end, the user repeats door opening and closing and sends base data to the server. Finally, the user activates the retired smartphone on. Fig. 2 shows illustration of the initialization phase.

2) Intrusion detection phase: Once our system is activated on, the sensors keep measuring and sending data to the server. The system consumes data from two on-board sensors: accelerometer and magnetometer. An accelerometer measures the rate of change of the velocity of an object (acceleration) with the unit of meters per second squared. It has enjoyed a variety of applications; especially it is useful for sensing vibrations or orientation in systems. In a modern smartphone, it measures acceleration on three axes to detect changes in orientation and to tell the phone. In this way, the smartphone knows itself up from down and rotates its own screen. A magnetometer measures magnetic fields along three perpendicular axes X, Y and Z. It produces voltage proportional to the strength and polarity of the field along the axis each sensor is directed. By varying its voltage output to a smartphone, it can tell the smartphone's orientation (rotation) relative to the Earth's north. The sensor reports data in the unit of micro Tesla [μT].

By using the sensor records, the server runs a detection algorithm developed based on *k*-Nearest Neighbors and Dynamic Time Warping algorithms to determine a door opening event. When an intruder tries to enter into home by opening the door, therefore, it is able to detect an abnormal event. If the data is deemed to be intrusive, the server sends a push alarm to the user, and the user can check video stream generated from a web camera installed at home. Fig. 3 demonstrate steps of the intrusion detection phase.

C. Background Knowledge

1) k-Nearest Neighbor (k-NN): *k*-NN is a machine learning algorithm for classification [15,16]. Once data is received, it searches k most similar data out of all the base data stored in it and determines which category the new data belongs to. Fig. 4 describes how it works.

The following is a simple example of the *k*-NN process where we assume that data progresses with two components [x, y].

a) Store base data in own database. Each base data is already classified based on own components x and y. Fig. 4(a) shows all the base data that is classified into two groups, A and B.

b) A new data (star marked) comes, and *k*-NN is asked to decide which group the data belongs to.

c) The *k*-NN algorithm first computes distances from the new data to all other data. A distance represents how similar the new data is to an existing base data (see Fig. 4(c)).

d) Select *k* most similar data (having short distance) out of all the base data. In Fig. 4(d), *k* is 3, and two of them are from group A and one from B. The new data is then classified into group A by the decision of the majority.

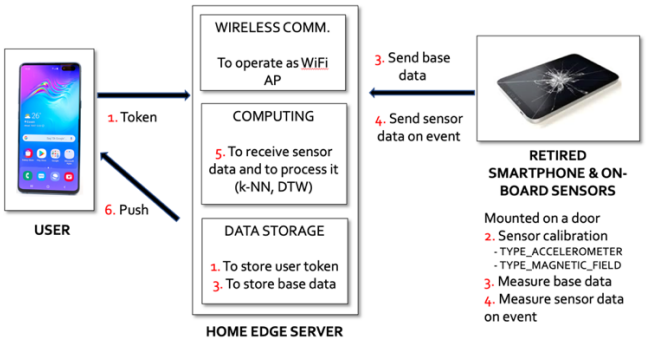


Fig. 1. The Proposed Home Intrusion Detection System Consists of a Retired Smartphone, a Recycled Home Edge Server, and a user. The Numbers, Red Colored, Represent Operation Steps.

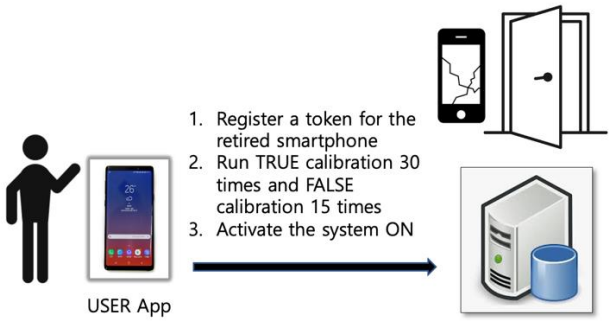


Fig. 2. An Initialization Phase Requires a Sensor Calibration Process.

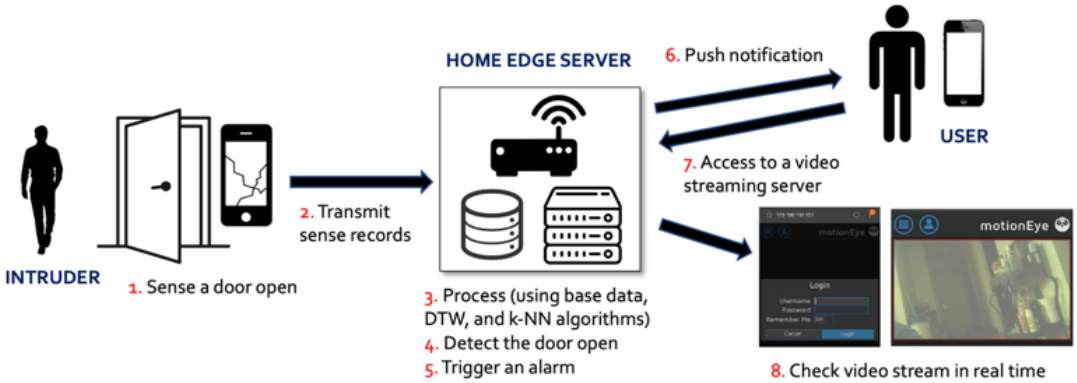


Fig. 3. The Detection Phase Senses an unauthorized Door Open, Pushes an Alarm Notification to a user, and Provides Video Stream in Real Time.

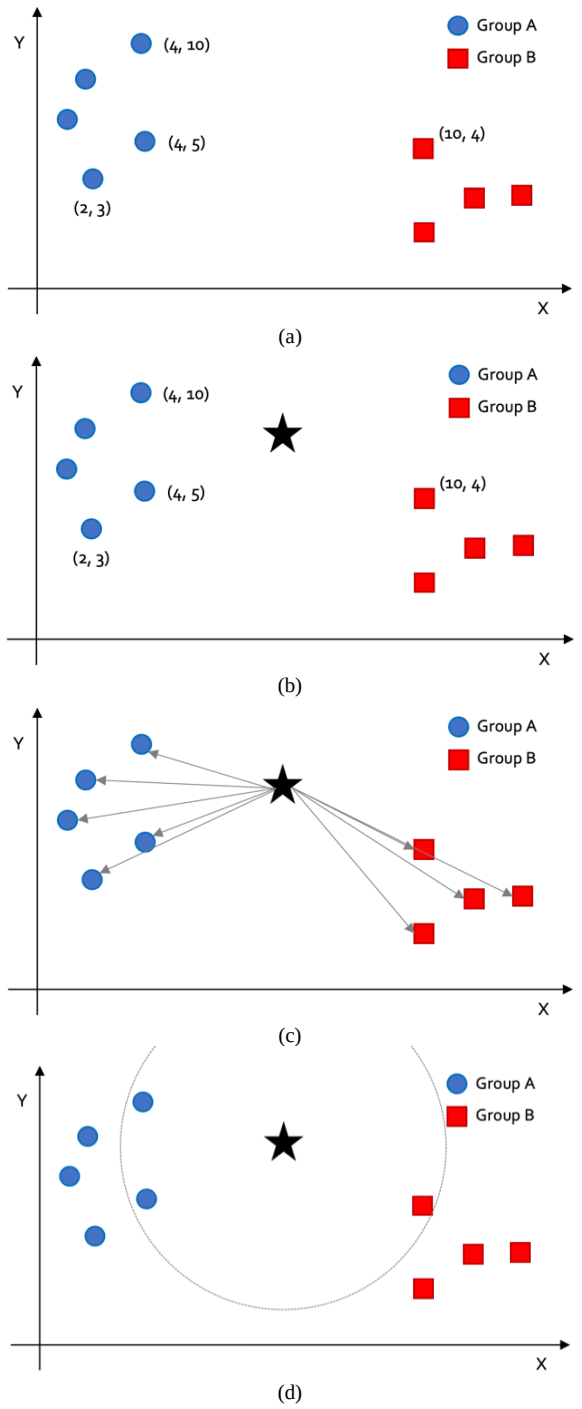


Fig. 4. Graphical Description of the Computation Process of *k*-NN Algorithm.

There are a few things to consider when applying *k*-NN. In the case of high-dimensional data, it is necessary to reduce the dimension because the curse of the dimension occurs when using the Euclidean distance formula. The proposed system measures the distance between high-dimensional data by using a Dynamic Time Warping algorithm without reducing the dimension. Therefore, the *k* value should be chosen as an odd number approximating the square root of the underlying data.

2) Dynamic Time Warping (DTW): DTW is an algorithm for determining similarity between time series data [17]. In addition to comparing the same temporal index as itself, the neighboring indexes are also compared to select more similar elements as their pairs. Below is an example illustrating the DTW process. Suppose two time-series data ***A*** and ***B***, each of which have six components as shown in Fig. 5.

DTW computes similarity of ***A*** and ***B*** in the following ways.

(i) Create a 6x6 matrix *M* and fill out all the elements with a predefined initial value, named max.

(ii) Set *M*(0,0) = | A0 – B0 |. In rows of the 1st column, i.e., *M*(1,0) ~ *M*(5,0), | Ax– B0 | is added to the previous row. For instance, *M*(1,0) = *M*(0,0) + | A1 – B0 | = 6 + | 2 – 7 | = 11. For columns of the 1st row, *M*(0,1) ~ *M*(0,5), | A0– Bx | is added to the previous column. *M*(0,1) is computed as *M*(0,0) + | A0– B1 | = 6 + | 1– 8 | = 13. In this way, the first rows and columns are filled out as shown in Fig. 6(a).

(iii) The rest of the elements, *M*(1,1) ~ *M*(5,5), is computed as follows. For *M*(i,j), pick three values, *M*(i-1,j-1), *M*(i-1,j), and *M*(i,j-1), and find the minimum (say, *M*(i,j,min)). Then, *M*(i,j) = *M*(i,j,min) + | Ai– Bj |. For instance, *M*(2,3) = *M*(2,3,min) + | A2– B3 | = *min*(13, 12, 17) + | 2– 5 | = 12 + 3 = 15. Fig. 6(b) shows outputs of this computation.

(iv) The last element, *M*(5,5) = 34, represents a DTW value. The last step routes back to the first element from this element. For *M*(i,j), it pick the minimum out of three elements, *M*(i-1,j-1), *M*(i-1,j), and *M*(i,j-1), and then jump to it. Repeat this until we reach to the first element. For instance, DTW is at *M*(5,5) at first. Then, it jumps to *M*(4,4), *M*(3,3), *M*(2,2), *M*(1,1), and *M*(0,0) sequentially as shown in Fig. 6(c). The distance is 5 in this example.

(v) Similarity between two data ***A*** and ***B*** is computed as DTW / distance = 34 / 5 = 6.8. The figure also shows that the route makes the main diagonal. This indicates that two data change constantly.

DTW in our system is used to derive similarity between newly received sensor data and underlying base data.

A	1	2	3	4	2	3
B	7	8	5	9	11	9

Fig. 5. An Example of DTW. This Assumes that there are Two Time Series Data, ***A*** and ***B***, each of which Contains 6 Components, [A0, A1, …, A5, B0, B1, …, B5].

6	6+7 (13)	13+4 (17)	17+8 (25)	25+10 (35)	35+8 (43)
6+5 (11)	max	max	max	max	max
11+4 (15)	max	max	max	max	max
15+3 (18)	max	max	max	max	max
18+5 (23)	max	max	max	max	max
23+4 (17)	max	max	max	max	max

(a)

6	6+7 (13)	13+4 (17)	17+8 (25)	25+10 (35)	35+8 (43)
6+5 (11)	6+6 (12)	12+3 (15)	15+7 (22)	22+9 (31)	31+7 (38)
11+4 (15)	11+5 (16)	12+2 (14)	14+6 (20)	20+8 (28)	28+6 (34)
15+3 (18)	19	15	19	26	31
18+5 (23)	24	18	22	28	33
23+4 (17)	28	20	24	30	34

(b)

6	6+7 (13)	13+4 (17)	17+8 (25)	25+10 (35)	35+8 (43)
6+5 (11)	6+6 (12)	12+3 (15)	15+7 (22)	22+9 (31)	31+7 (38)
11+4 (15)	11+5 (16)	12+2 (14)	14+6 (20)	20+8 (28)	28+6 (34)
15+3 (18)	19	15	19	26	31
18+5 (23)	24	18	22	28	33
23+4 (17)	28	20	24	30	34

(c)

Fig. 6. DTW Computation makes a Matrix and Calculates Elements from Time Series Data.

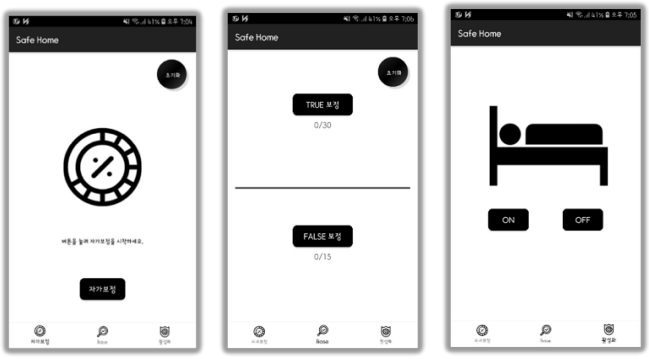
IV. Development

This section describes our development of the proposed intrusion detection system.

A. Retired Smartphone and On-Board Sensors

A retired smartphone, mounted to a door, serves as a group of sensors. That is, it senses events and sends measurement data to the server. We use Samsung SM-G900L that is with CPU of Qualcomm Snapdragon 2.5Ghz Quad-Core Krait 400, GPU of Qualcomm Adreno 330 578Mhz, and 2GB of memory [18]. It runs on Android 6.0.1 and API 23 versions. Out of a full list of on-board sensors, our system makes use of an accelerometer and a magnetometer. In order to control the retired smartphone and sensors, we develop a mobile application as shown in Fig. 7.

Using the application, a user is able to perform the initialization phase. First, a self-calibration measures values from the accelerometer and the magnetometer for 5 seconds when a door is closed and computes average values (see Fig. 7(a)). These values are used as references in a general situation (i.e., *stationary state*); they are stored in the x, y, and z reference variables of each sensor and are used in our detection algorithm to determine future events. We note that x, y, and z are from the accelerometer and represents three motion axes (components) related to the orientation of the retired smartphone. Next, the application starts a TRUE/FALSE calibration process (Fig. 7(b)). When the door is opened through a TRUE correction button, an array of 150 data is transmitted to the server 30 times, each of 50 x, y, and z coordinates. Thereafter, a FALSE correction button transmits an array of 150 data to the server 15 times. These values are stored in the server and used as reference when judging whether the door is on *intrusion state* (i.e., door opening) or on *noise state* (e.g., door knocking). Last, the user is able to activate and deactivate the retired smartphone ON and OFF through the buttons. Because sensor measurements during activation take place in the background service, it works even if the screen of the smartphone is turned off after installation is complete. The accelerometer and the magnetometer start sensing when activated. The retired smartphone sends accelerometer values to the server when an event occurs based on the mean value of a magnetic force value.



(a) Self-calibration (b) Sending base data (c) Activation

Fig. 7. A user makes an Initial Setup of the Retired Smartphone via a Mobile Application.

B. Recycled Home Edge Server

One of the keywords in our research is recycling; that is, our home server can be developed using existing home devices. For instance, a Wi-Fi router can be modified with OpenWrt [19] and support computing resources. Alternatively, an old PC at home can be used. Our system develops a home server using Raspberry Pi 3 Model B+ that is on Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz with 1GB memory [20]. The server provides three main functions; Wi-Fi access point, edge computing, and local database.

1) Wi-Fi Access Point (AP Mode): The server is configured so as to serve as a Wi-Fi access point, Access Point (AP) mode [21]. Thus, it provides an Internet connectivity service to all the home devices. With this configuration, the server is able to make a direct communication with the retired smartphone, which reduces transmission delay.

2) Data Processing for Intrusion Detection (Edge Computing Mode): Our server provides edge computing capability [2,3]. Because of performance and privacy issues, the edge computing is becoming a new computing model. Since it is not easy for non-IT people to use cloud computing, moreover, it is quite important to develop a fully localized system. In our system, the retired smartphone transmits raw data without pre-processing to the server. Instead, the server is responsible for data processing for intrusion detection, which helps save battery life of the retired smartphone.

The server converts a registration token received from the user and base data sent from the retired smartphone into a string and stores it in Redis database [22]. Upon receiving accelerometer data from the retired smartphone, it runs the *k*-NN algorithm to determine whether it is noise or meaningful data implying a door is actually opened. Noise can be either natural noise or environmental noise. Fine vibrations that a human being notices hardly are called natural noise, whereas impacts on doors such as knocking and vibrations from surrounding objects generate environmental noise. More technically, the retired smartphone computes the absolute value of difference between sampling data xN, yN, zN out of raw data and x, y, and z values obtained by self-calibration process at the initialization phase. Output of this computation, [x1, x2, x3, … , y1, y2, y3, ... , z1, z2, z3, ...], is then transmitted to the server. Then, the server normalizes both all 45 sets of base data and the received data; a random value X is divided by the maximum value out of the data, multiplied by 2, and minus 1. The server computes distance (similarity) between base data and the received data using the DTW algorithm. It then sorts the derived similarities in ascending order and select k values that are similar (the closer to zero, the similar). The majority vote is performed on the selected *k* value to determine whether the received data is noise or represents a door opening. When the reading results are not noise, the user is notified via push notification.

C. Web Video Service: Video Streaming in Real Time

Our system also implements a web video service by connecting a camera to the server and by using MotionEye open source library [23]. With this feature, a user can enjoy video streaming in real time by accessing the server as shown in Fig 8. For the convenience of service, the user can check for intrusions once receiving a push alarm. The library also offers a variety of functions including motion detection and storage. When using a motion detection function, the server automatically saves a video clip to a file upon detecting movement. Therefore, depending on a user's needs, a variety of functions can be set, such as setting images or storing images.

D. User

Our system also implements a simple mobile application that can be controlled by a user. By clicking ON or OFF button in the application, the user can easily register or unregister herself to the home server. On registration, a token is generated and transmitted to the server. Push notification can be received through the server when a door is opened upon activation. A camera button allows the user to access the web streaming server and watch real-time video.

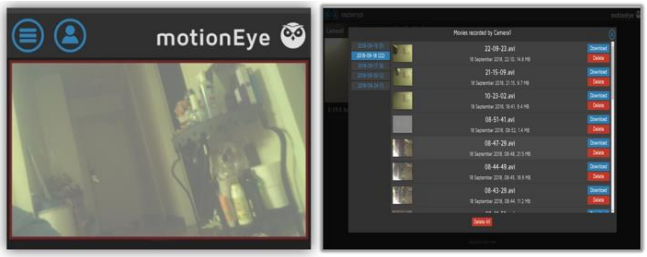


Fig. 8. A user can Double-Check Situation at the Door via Video Stream.

V. Experiments and Performance Evaluation

An eventual goal of this section is to evaluate detection accuracy of the proposed system. For experiments, the retired smartphone is mounted near a handle on the steel door of a classroom, as shown in Fig. 9, so that on-board sensors can spin when the door is opening. The first experiment measured base data that was then used as a training set. The next experiment recorded sensor data under various conditions, processed it using the proposed detection algorithm, and computed detection accuracy of the system.

Fig. 9. In our Experiment, the Retired Smartphone is Mounted on a Steel Door near the Handle.

A. Measurement of Base Data

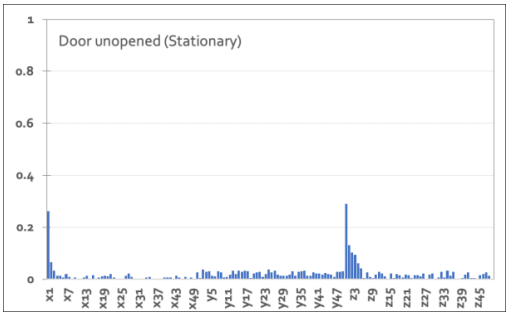
The first experiment collected base data (i.e., ground truth data sets) that was classified into to two groups – TRUE event (the door is opening or intrusion state) and FALSE event (stationary state and noise state). All the base data included 30 TRUE events and 20 FALSE events. TRUE data represented 10 events of the door opening slowly, 10 events of the door opening at a normal speed, and 10 events of the door opening quickly. FALSE data represented 10 events of the door remained stationary (unopened) and 10 events of door knocking. We also ran experiments on a glass door that recorded 20 FALSE data; 10 events with a gentle wind and 10 events with the door shaking heavily.

Fig. 10(a) draws sensor records collected from the accelerometer when the door remains stationary. X-axis represents normalized sampling data, and Y-axis indicates acceleration values in the unit of meter per second per second. We note that all the graphs in this subsection are drawn on the same X-Y plane. Most values in the figure remain zero, representing tiny vibrations on the door. Two spikes are

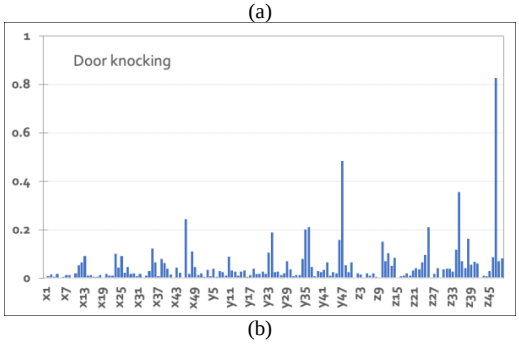
measured; further investigation found that people were passing by the door at the moment. Fig. 10(b) shows acceleration values in the event of door knocking. Vibration is measured each time we knock the door. In the experiment, we knocked the door 4 times with high strength that were captured by 4 spikes at each axis. The smallest (the first) spike occurred when knocking at the point 1 meter away from the smartphone, whereas the biggest (the last) one when 10 cm away.

Fig. 11 shows raw data collected from the accelerometer when the door were opening at three different speeds. When the door were opening slowly, we observed that values at the X and Y samples were ignorable as shown in Fig. 11(a). Their average values are 0.022 and 0.021, respectively. But, variation at the Z samples is noticeable, up to 0.228. When speeding up the door opening, the values at the Z samples become higher; almost 5 times bigger at the peak, 1.023, than that at a slow speed (Fig. 11(b)). Unlike the Y samples, values at the X samples increase gradually and go up to 0.41 at max. This indicates that the door vibrates slightly from side to side. This movement becomes quite remarkable when opening the door at a high speed (Fig. 11(c)). The maximum value reaches to 1.057. We also note that values at the Z samples double and draw the same shape when comparing to those at the normal speed.

Our experiments also measured another 20 FALSE data, in which we set up a wobbly commercial glass door. The first experiment recorded sensor values on a gentle wind, and its results are demonstrated in Fig. 12(a). Values range from 0 to 0.09 that are under an internal threshold. When the door shook heavily, values at the Z samples fluctuated accordingly (Fig. 12(b). While they goes over 1.6 on the Y-axis, the up-and-down pattern is quite different from them in the case of door opening. The proposed detection algorithm is able to distinguish the patterns after a learning phase.

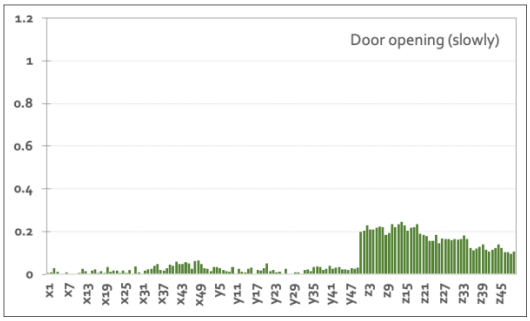


(a)

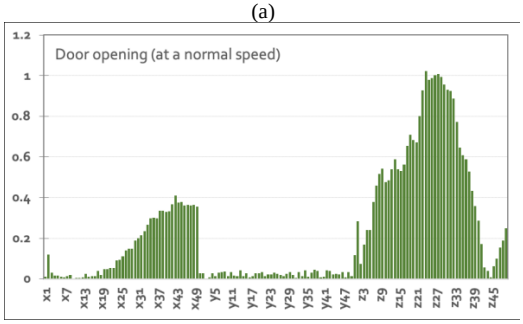


(b)

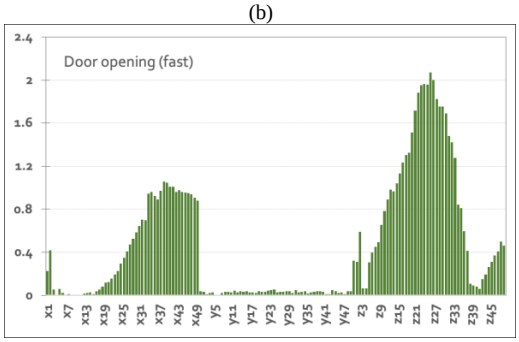
Fig. 10. Representation of Raw Data from the Accelerometer when a Door Remains Stationary (a) and when a Door is Knocked (b).



(a)

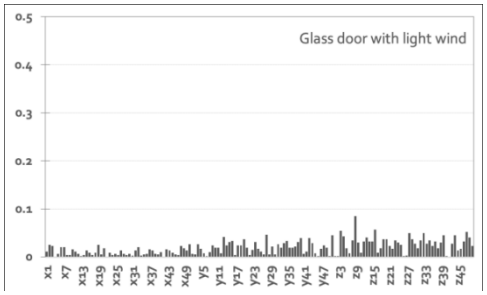


(b)

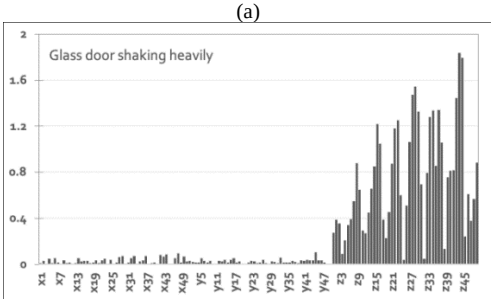


(c)

Fig. 11. Representation of Raw Data from the Accelerometer when a Door is Opening at a Slow Speed (a), A Moderate Speed (b), and A High Speed (c).



(a)



(b)

Fig. 12. Representation of Raw Data from the Accelerometer when there is a Light Wind (a) and when a Door is Shaking Heavily (b) on a Glass Door.

B. Evaluation of Accuracy of the Proposed System

This subsection describes our experiments to evaluate that the proposed system detects door opening under various conditions. We (i) knocked the steel door multiple times lightly and heavily, (ii) opened the door at 2 and 5 seconds of speeds, and (iii) approached a magnet to the door. The proposed detection algorithm made use of on-board sensors differently; (i) using accelerometer data only, (ii) using magnetometer data only, and (iii) using both of them together. We repeated the experiments both on a wooden door and on a glass door. The experiments were conducted 20 times each by producing various situations, and results are summarized in this subsection.

1) Using accelerometer data only: The detection algorithm in the first experiment that collected the accelerometer data only and processed it using two internal algorithms, *k*-NN and DTW. Its result of detection accuracy is shown at the first three columns from the left in Table I. *True Positive (TP)* represents the state that the proposed detection algorithm recognizes the door opening correctly, whereas *True Negative (TN)* implies the state that the algorithm correctly classifies input data as noise or recognizes that the door remains stationary.

When the door was knocked lightly, one event were falsely recognized as detection. As the strength of door knocking increased, the number of false detection also increased; 85% of detection accuracy with heavy knocking. Further investigation found that all the false detection occurred when knocking the door very fast. This condition drew a curve of sensor samples similar to that when opening a door slowly. When the door was opening at a moderate speed for about two seconds, detection accuracy recorded 70% with 14 correction detection and 6 false detection. When slowing down the speed (about five seconds), the number of false detection increased up to 8 with detection accuracy of 60%. The results indicate that the accuracy of door opening is quite low when accelerometer data is used only. In particular, when opening a door slowly Z-axis data was only sampled from the sensor and its values were small as demonstrated in the previous subsection. Therefore, the detection algorithm is likely to recognize the situation as noise with high probability. In the case of a magnet approaching, 100% detection accuracy was shown because the magnet did not affect movement of the door. Thus, the sensor did not capture meaningful data for detection of door opening.

2) Using magnetometer data only: The next three columns in the table represent results of the second experiment that uses the magnetometer data only in the detection algorithm. Detection accuracy in the cases of knocking and door opening shows 100%. These results are quite straightforward. As noted, the sensor can tell a device's rotation relative to the magnetic north, and here, the device is the door. Therefore, it can detect the door's movement accurately regardless of opening speeds, which results in high detection rate. However, it should be noted that the magnetometer also reacts to a magnet. When we approached a magnet to the door and manipulated it properly, the detection algorithm falsely recognized all the events as detection. Thus, the table shows 0% of accuracy. In summary, while magnetometer data is able to detect intrusion with high probability, it is not recommended to use it only because of critical weakness.

3) Using data from both accelerometer and magnetometer: Previous experiments show that it is not possible to detect intrusion correctly when using data from a single sensor. At the third experiment, thus, the detection algorithm makes use of data both from the accelerometer and the magnetometer. The last three columns in the table summarizes the experimental results.

In the case of door knocking, the strength of knocking generated tiny vibration on the door. Moreover, a small rate of false detection was corrected by the magnetometer. In this way, detection accuracy achieves 100%. In the case of door opening, detection performance improves remarkably, comparing to results from the experiment using the accelerometer data only; from 70% to 100% in slow opening and from 60% to 95% in fast opening. These values indicate that one can advance performance of a system by coordinating more data in general. But, we note that using many sensors and heterogeneous data from them may make a system complicated and introduce unwanted processing delay. Therefore, a special attention must be payed to. In the experiment of magnet approaching, detection accuracy of 100% was achieved. This result is mainly attributed to the role of the accelerometer; a magnet did not affect movement of the door, and thus the detection algorithm successfully recognized that the door was not opened.

4) Various door materials: The last experiment aims to evaluate performance of our detection algorithm on a wooden door and on a glass door in a residential building. To this end, we mounted the retired smartphone to them and set up the detection algorithm to use data from both the accelerometer and the magnetometer. The experiment ran three situations: door knocking, door opening, and magnet approaching.

Table II summarizes experimental results. In the experiments of door knocking (the first two rows from the top in the table), the algorithm recognized all the events correctly, resulting in 100% of accuracy on both materials. In the experiment of quick door opening, it also detected the events with 100% of probability. However, when opening the door slowly, 3 and 4 false detection were observed on the wooden and the glass doors, respectively. The 85% and 80% of accuracy are mainly attributed to the speed of door opening. Since the door was opened physically, the magnetometer sensed it clearly. The accelerometer recorded small values of acceleration because of the slow speed, which led our detection algorithm to classify them into noise.

TABLE I. DETECTION ACCURACY (ACC, %) IN DIFFERENT SETTING UNDER VARIOUS SITUATIONS. TP: TRUE POSITIVE AND TN: TRUE NEGATIVE

		Using accelerometer data only			**Using magnetometer data only**			**Using both data**		
		TP	***TN***	***Acc***	***TP***	***TN***	***Acc***	***TP***	***TN***	***Acc***
Knocking (multiple)	***lightly***	1	19	0.95	0	20	1.0	0	20	1.0
	heavily	3	17	0.85	0	20	1.0	0	20	1.0
Door opening	***2 sec of speed***	14	6	0.7	20	0	1.0	20	0	1.0
	5 sec of speed	12	8	0.6	20	0	1.0	19	1	0.95
Magnet approaching		0	20	1.0	20	0	0	0	20	1.0

TABLE II. DETECTION ACCURACY (ACC, %) ON DIFFERENT DOOR MATERIALS. TP: TRUE POSITIVE AND TN: TRUE NEGATIVE

Using both accelerometer and magnetometer data		**On a wooden door**			**On a glass door**		
		TP	***TN***	***Acc***	***TP***	***TN***	***Acc***
Door knocking (multiple)	***lightly***	0	20	1.0	0	20	1.0
	heavily	0	20	1.0	0	20	1.0
Door opening	***2 sec of speed***	20	0	1.0	20	0	1.0
	5 sec of speed	17	3	0.85	16	4	0.8
Magnet approaching		0	20	1.0	0	20	1.0

5) Discussion: Experimental results showed that detection accuracy reached up to 99% on average. This results can be compared to average accuracy of 93%~97% in previous works. This confirms that better performance is attributed to more data from multiple on-board sensors. We note that our experiments included a scenario that a door was opening at a very slow speed. Such slow movement could make some sensors hard to capture the door event. Since previous research never considered the scenario, it is not possible to compare performance directly. Our system could detect the event with high probability mainly thanks to using multiple sensors. With this approach, however, the retired smartphone generates/processes more data and consumes more energy. By offloading all the computing processes to a local home edge server, our system could reduce overhead on the smartphone.

Our experiment on various door materials is another highlight that may attract readers' interests. A primary role of a sensor in the emerging Internet of Things (IoT) is to capture phenomena (or events) occurring in physical environment. Different door materials represent different environment. In this sense, the sensor may recognize the same event (i.e., door opening) differently once it is on different physical materials. IoT research of future may want to consider such subtle interactions between sensors and physical environment of interests.

We also believe that our solution shows feasibility of recycling IT devices to our society as well as enables a home owner to build own security system with no cost. Recycling retired, existing devices enables us to save cost for disposing old devices and to make the best of surplus IT resources, otherwise wasted, on devices deployed widely today.

VI. CONCLUSION

This paper proposes a home intrusion detection system that is composed of recycled IT devices. It mounted a retired smartphone on an entrance door and used particular on-board sensors to records signals upon unwanted door opening. A Wi-Fi access point was reconfigured to serve as a home server, and thus it processed sensor data to detect unauthorized access to home by an intruder. The proposed system faced three technical challenges: inaccurate sensor data, detection of an event of interest out of the flood of big data, and energy consumption of the battery-powered smartphone. To solve the first two challenges, this paper proposed a lightweight detection method that employed the *k*-nearest neighbors algorithm and the dynamic time warping algorithm. The last challenge was resolved by using only two sensors and by offloading computation to the home edge server.

We developed a prototype where Samsung SM-G900L was reused and the server was implemented using a Raspberry Pi that was also equipped with a web camera. The first experiment measured base data to understand accuracy of on-board sensors in the smartphone as well as to calibrate sensor data. The next experiment considered three scenarios: (i) knocking a steel door multiple times lightly and heavily, (ii) opening the door at 2 and 5 seconds of speeds, and (iii) approaching a magnet to the door. It also set three options on the proposed detection algorithm. Experimental results demonstrated that detection accuracy could reach 95%-100% when the algorithm used data from an accelerometer and a magnetometer. The last experiment showed that our system was able to detection intrusion with high probability when using various doors of different materials.

REFERENCES

[1] Webcams vulnerable to attack. Available online: https://blog. malwarebytes.com/hacking-2/2019/09/15000-webcams-vulnerable-how-to-protect-webcam-hacking/ [accessed on 7/24/2020].

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, " Fog computing and its role in the internet of things," in Proceedings of ACM workshop on Mobile Cloud Computing, Helsinki, Finland, August 2012.

[3] E. Hamilton, "What is Edge Computing: The Network Edge Explained," 2018. Available online: https://www.cloudwards.net/what-is-edge-computing/ [accessed on 7/24/2020].

[4] M. Wu, P. Pathak, and P. Mohapatra, "Monitoring Building Door Events using Barometer Sensor in Smartphones," in Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp), Osaka, Japan, September 2015.

[5] T. Dissanayake, T. Maekawa, D. Amagata, and T. Hara, "Detecting Door Events Using a Smartphone via Active Sound Sensing," Proceedings of ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies., vol. 2, no. 4, article no. 160, 2018.

[6] M. Mahler, Q. Li, and A. Li. "SecureHouse - A Home Security System Based on Smartphone Sensors," in Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom), Hawaii, US., March 2017.

[7] L. Gong, Y. Zhao, C. Xiang, Z. Li, C. Qian, and P. Yang, "Robust Light-Weight Magnetic-Based Door Event Detection with Smartphones," in IEEE Transactions on Mobile Computing., vol. 18, no. 11, pp. 2631-2646, 2019.

[8] R. Behringer, M. Ramachandran, and V. Chang, "A Low-Cost Intelligent Car Break-In Alert System Using Smartphone Accelerometers for Detecting Vehicle Break-Ins," in Proceedings of International Conference on Internet of Things and Big Data, Rome, Italy, April 2016.

[9] K. Ohara, T. Maekawa, and Y. Matsushita, "Detecting State Changes of Indoor Everyday Objects using Wi-Fi Channel State Information," in Proceedings of ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies., vol. 1, no. 3, article no. 88, 2017.

[10] S. Shi, S. Sigg, and Y. Ji, "Passive detection of situations from ambient FM-radio signals," in Proceedings of ACM Conference on Ubiquitous Computing (UbiComp), Pittsburgh, US., September 2012.

[11] T. Hnat, E. Griffiths, R. Dawson and K. Whitehouse, "Doorjamb: Unobtrusive room-level tracking of people in homes using doorway sensors," in Proceedings of ACM Conference on Embedded Network Sensor Systems (SenSys), Toronto, Canada, November 2012.

[12] W. Jabbar, T. K. Kian, R. Ramli, S. Zubir, N. Zamrizaman, M. Balfaqih, V. Shepelev, and S. Alharbi, "Design and fabrication of smart home with internet of things enabled automation system," in IEEE Access, vol. 7, pp. 144059-144074, 2019.

[13] U. Ozeer, L. Letondeur, F.-G. Ottogalli, G. Salaün and J.-M. Vincent, "Designing and implementing resilient IoT applications in the fog: A smart home use case," in Proceedings of Conference on Innovation in Clouds, Internet and Networks, Paris, France, February 2019.

[14] S. Kaur, R. Singh, N. Khairwal and P. Jain, "Home automation and security system", in Advanced Computational Intelligence: An International Journal (ACII), vol. 3, no. 3, pp. 17-23, 2016.

[15] T. Cover and P. Hart, "Nearest neighbor pattern classification," in IEEE Transactions on Information Theory., vol. 13, no. 1, pp. 21-27, 1967.

[16] N. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," in The American Statistician., vol. 46, no. 3, pp. 175–185, 1992.

[17] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," in IEEE Transactions on Acoustics, Speech, and Signal Processing., vol. 26, no. 1, pp. 43–49, 1978.

[18] Samsung Galaxy S5 (SM-G900L) Specification. Available online: https://www.phonemore.com/specs/samsung/galaxy-s5/sm-g900l/ [accessed on 7/24/2020].

[19] The OpenWrt Project. Available online: https://openwrt.org/ [accessed on 7/24/2020].

[20] Raspberry Pi 3 Model B+. Available online: https://www. raspberrypi.org/products/raspberry-pi-3-model-b-plus/ [accessed on 7/24/2020].

[21] Setting up a Raspberry Pi as a routed wireless access point. Available online: https://www.raspberrypi.org/documentation/configuration/wireless/access-point-routed.md [accessed on 7/24/2020].

[22] Redis. Available online: https://redis.io/ [accessed on 7/24/2020].

[23] MotionEye. Available online: https://github.com/ccrisan/motioneye/wiki [accessed on 7/24/2020].